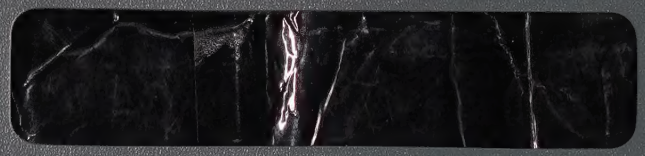
SYSTRON DONNER

A THORN EMI company



INSTRUCTION MANUAL

3520

MODEL 3520
INSTRUMENTATION CONTROLLER

Copyright © 1980 Systron-Donner Corporation

SYSTRON DONNER CORPORATION

CONCORD INSTRUMENT DIVISION
2727 Systron Drive • Concord, California 94518 • U.S.A.
Tel: (415) 676-5000 • TWX: 910-481-9479 • Cable: SYSTRONDONNER
FSCM No. 52542

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_05206

SYSTRON DONNER
Member THORN EMI Group

INSTRUMENT
DIVISION

2727 Systron Drive, Concord, CA 94518
Phone (415) 676-5000
TWX 910-481-9479

PROPRIETARY RIGHTS STATEMENT

The design information contained herein is the sole property of Systron Donner. The disclosure of this information does not constitute the release of proprietary rights thereof, but is to be used for informational purposes only. Permission to reproduce this information of the product described herein must be obtained in writing from Systron Donner.

Copyright © 1982 Systron Donner
Printed in U.S.A.

SYSTRON DONNER
INSTRUMENT DIVISION
2727 Systron Drive
Concord, California 94518

WARRANTY POLICY

I Warranty

Systron Donner Instrument Division warrants that at the time of shipment the products manufactured by Systron Donner and sold hereunder will be free from defects in material and workmanship and will conform to the current specifications published by Systron Donner's Instrument Division.

II Warranty

A The warranty period will be one year from the date of shipment. This warranty is extended only to the original buyer.

III Warranty Adjustments

A. If any defect within this warranty appears, Purchaser shall notify Seller immediately.

B Systron Donner agrees to repair or furnish a replacement for, but not install, any product which within the standard warranty shall, upon test and examination by Systron Donner, prove defective within the above warranty period.

C No product will be accepted for return or replacement without prior approval of Systron Donner. Upon such authorization and in accordance with instructions of Systron Donner, the product will be returned to the manufacturing location, shipping charges prepaid by Purchaser.

IV Exclusions from Warranty

A The foregoing warranty is in lieu of and excludes all other expressed or implied warranties of merchantability or fitness or otherwise.

B Systron Donner will not be liable for any special or consequential damages or for loss, damages or expense directly or indirectly arising from the use of the products, or any inability to use them either separately or in combination with any other equipment or material, or from any other cause.

C The warranty does not extend to any product manufactured by Systron Donner which has been subjected to misuse, neglect, accident, improper installation or to use in violation of instructions furnished by Systron Donner.

D The warranty does not extend to or apply to any unit which has been repaired or altered at any place other than Systron Donner; by persons not expressly approved by Systron Donner; not to any unit, the serial number of which has been removed or defaced or Changed.

E This warranty applies to new equipment only and will cover repaired or replacement items only to the extent of the balance remaining of the items original warranty period noted in Paragraph ll.

F All freight charges will be paid for by the Purchaser.

June, 1985

TABLE OF CONTENTS

CHAPTER 1 - LEARNING TO PROGRAM YOUR BUSSER PAGE

1.1 Introduction 1-1
1.2 Specifications 1-2
1.3 Receiving Your BUSSER 1-4
1.4 Power Up 1-4
1.5 Exploring the Keyboard 1-4
1.5.1 Typewriter Keys 1-4
1.5.2 Using the Shift Key 1-4
1.5.3 Using the Character Delete Key 1-5
1.5.4 Program Control Keys 1-6
1.5.5 BUS Control Keys 1-6
1.5.6 ROM Control Keys 1-6
1.6 Programming the BUSSER 1-7
1.6.1 Using the Return Key 1-7
1.6.2 Error Messages 1-7
1.6.3 Using Print with Strings 1-8
1.6.4 Using Print with Variables 1-8
1.6.5 Programming 1-9
1.6.6 Immediate and Deferred Modes 1-9
1.6.7 Creating a 'GAS BILL' Program 1-9
1.6.8 Using the Semicolon 1-10
1.6.9 Variables 1-10
1.6.10 Assignment 1-11
1.6.11 Using GOTO 1-11
1.6.12 Using LIST 1-12
1.6.13 Using STEP 1-12
1.6.14 Correcting Entire Lines 1-12
1.6.15 Running the Program 1-13
1.6.16 Using Stop in Immediate Mode 1-13
1.6.17 Using Stop in Deferred Mode 1-14
1.6.18 Amending the Program 1-14
1.6.19 Recording the Program 1-15
1.6.20 Loading the Program 1-15
1.6.21 The PROM Cartridge 1-16
1.6.22 Self Test 1-16
1.6.23 Autostart 1-17
1.6.24 PROM Control Using Autostart 1-18
1.7 Programming with IEEE Instruments 1-19
1.7.1 IEEE Programming 1-19
1.7.2 Connecting the IEEE Instrument 1-19
1.7.3 Using BUSCLEAR 1-20
1.7.4 Using BUSREM 1-21
1.7.5 Using BUSADDRESS 1-21
1.7.6 Using BUSOUT 1-22
1.7.7 Using For - Next Loops 1-24
1.7.8 Using BUSPRINT 1-27
1.7.9 Using PRINT in an IEEE Program 1-28
1.7.10 Using NEXT 1-28
1.7.11 Listing the Program 1-29

CHAPTER 1 - LEARNING TO PROGRAM YOUR BUSSER (Cont'd) PAGE

1.7.12 Running the Program 1-30
1.7.13 Auto-Verbs 1-30

CHAPTER 2 - COMMAND DICTIONARY

2.1.1 Program Control Keys 2-1
2.1.2 BUS Control Keys 2-2
2.1.3 ROM Control Keys 2-4
2.1.4 High Level Language 2-5
2.1.4.1 Assignment 2-5
2.1.4.2 Arrays 2-5
2.1.4.3 Programming Branching 2-5
2.1.4.4 Iteration 2-6
2.1.5 Arithmetic Operators 2-6
2.1.6 Numerical Decisions 2-6
2.1.7 Symbols 2-7
2.1.8 Miscellaneous 2-7
2.1.9 IEEE Control Codes 2-8
2.2 Examples 2-9
2.2.1 STEP Mode 2-9
2.2.2 Service Request 2-11
2.2.3 Serial Poll 2-11
2.2.4 BUSIN 2-12
2.2.5 BUSPRINT 2-13
2.2.6 Iteration 2-14
2.2.7 Numerical Operators 2-15
2.2.8 Numerical Decisions 2-16
2.2.9 IEEE BUS Buffers 2-17
2.2.10 Tabulation 2-19
2.2.11 Cart-Info 2-19
2.2.12 Secondary Addressing 2-19
2.2.13 Bustime Specification 2-20
2.2.14 Program Chaining 2-20
2.2.15 PROM Eraser Accessory 2-21
2.3 Error Dictionary 2-22
2.3.1 Program Error Messages 2-22
2.3.2 IEEE Buffer Errors 2-23
2.3.3 Bus Error Messages 2-23
2.3.4 ROM Error Messages 2-23

CHAPTER 3 - REPLACEABLE PARTS (Cont'd) PAGE

Interconnect Cable Assembly #07363601 3-12
A3, Display PC Assembly #07362301 3-13
A4, Processor PC Assembly #07362401 3-14
Option 001, PROM 2k Reader/Programmer #07366801 3-16
Option 001, PROM 2k Reader/Programmer #07364301 3-17
Option 001, PROM Module 2k Assembly #07363001 3-18
Option 001, 2k PROM Module PC Assembly #07363701 3-19
Option 001, PROM Module 4k Assembly #07363002 3-20
Option 001, PROM Module 4k PC Assembly #07369802 3-21
Control Line Status Indicator Assembly #07366301 3-22
Circuit Board PC Assembly #07369901 3-23
PROM Eraser Assembly #073701 3-24
PROM Eraser PC Assembly #07367501 3-25

CHAPTER 4 - DRAWINGS

4.1 Introduction 4-1

LIST OF FIGURES

Figure 1-1 Teletype Keyboard 1-5
1-2 Program Control Keys 1-6
1-3 Bus and ROM Control Keys 1-7
2-1 IEEE Statements 2-10
4-1 Model 3520 Final Assembly #07362001 4-3
4-2 Model 3520 Chassis Assembly #07362101 4-4
4-3 A1, Power Supply Assembly #07363401 4-5
4-4 A1, Power Supply Schematic #7-07363401 4-5
4-5 A2, Keyboard PC Assembly #07364001 4-7
4-6 A2, Keyboard Schematic (Sheet 1 of 2) #7-07364001 4-8
4-6 A2, Keyboard Schematic (Sheet 2 of 2) #7-07364001 4-9
4-7 A3, Display PC Assembly #07362301 4-10
4-8 A3, Display Schematic #7-07362301 4-11
4-9 A4, Processor PC Assembly #07362401 4-12
4-10 A4, Processor Schematic #7-07362401 4-13
4-11 Option 001, 2k/4k PROM Module Assembly #07363001 4-14
4-12 Option 001, 2k PROM Module PC Assembly #07369801 4-14
4-13 Option 001, 2k PROM Module Schematic #7-07363001 4-15
4-14 Option 001, 4k PROM Module PC Assembly #07369802 4-16
4-15 Option 001, 4k PROM Module Schematic #7-07363002 4-17
4-16 Control Line Status Indicator Assembly #07366301 4-18
4-17 Control Line Status Indicator PC Assembly #07369901 4-19
4-18 Control Line Status Indicator Schematic #7-073663 4-21
4-19 PROM Eraser Assembly #073701 4-22
4-20 PROM Eraser PC Assembly #07367501 4-22
4-21 PROM Eraser Schematic #7-073701 4-23

LIST OF TABLES

			PAGE
Table	1-1	Specifications	1-1
	1-2	Voltage Entry Characters	1-22
	1-3	Range Entry Characters	1-23
	1-4	Polarity Entry Characters	1-23
	1-5	IEEE-488 Programming Format Meaning	1-24
	1-6	Auto-Verbs	1-30
	3-1	Manufacturer's Code-To-Name Index	3-2
	4-1	List of Drawings	4-1

APPENDIX

A	Binary and Decimal Code to ASCII Character	A-1
B	Multiline Interface Messages: ISO Code Representation	B-1

1.2 SPECIFICATIONS

Table 1.1 lists the specifications applicable to the Model 3520 BUSSER.

TABLE 1.1 SPECIFICATIONS

DISPLAY

- No. of lines: 1
- Characters per line: 40
- Character Set: 64 upper case ASCII characters, each formed by a 5 x 7 dot matrix.
- Character Size: 0.140" wide by 0.200" high

I/O INTERFACE

- IEEE standard 488 - 1978 Interface

COMPUTER SYSTEM

- Type: 6802 Microprocessor
- Internal Storage: 6K Bytes, ROM; 2K Bytes, RAM
- Numeric Range: ±32767 (integer)

KEYBOARD AND PANEL CONTROLS

- Standard Teletypewriter Layout
- Special Function Keys

SOFTWARE SYSTEM

- SysDL I Language statements and commands
- IEEE Bus Driver Commands
- Arithmetic Operators
- Relational Operators
- Two Dimensional Arrays
- Subroutines and Nested Subroutines
- Single Step Operation
- Immediate/Deferred Modes of Operation
- Error Messages

CHAPTER 1
LEARNING TO PROGRAM YOUR BUSSER

1.1 INTRODUCTION

Congratulations on your purchase of the Systron-Donner "BUSSER" Model 3520. The BUSSER fills a long-standing need for an IEEE bus controller with an outstanding price/performance ratio. The BUSSER combines the power of single-stroke keys with an easy-to-understand version of the BASIC programming language.

Your BUSSER may be a natural choice for instrument checkout, incoming inspection, service repair, instrument demonstration and quality control. However, the BUSSER may also fill a need as a controller in a small systems environment.

How can the BUSSER serve so many purposes? The features speak for themselves. Routine IEEE bus commands, program storage, and program control commands are all handled by single-stroke function keys. The remainder of the keyboard is laid out in a traditional standard TTY format.

The programming itself is performed with an easy-to-learn version of the high level BASIC language. A versatile single step mode handily allows the display of each program step prior to final execution as well as facilitating many other programming tasks. Frequently used programs are easily stored in the plug-in, U.V. erasable PROM.

The handiest feature of the BUSSER is the price. Solutions to many of your IEEE bus problems are now within reach at a reasonable cost.

In order to best learn to operate the BUSSER, please use this manual as a teaching guide. Start at the beginning of the following text and use your BUSSER and this manual side-by-side. By the time you are done with Chapter 1, you will be well on your way to using your BUSSER to its fullest advantage.

TABLE 1.1 SPECIFICATIONS (Cont'd)

GENERAL

Physical Characteristics: 13 1/2" wide x 13 1/2" deep x 5 1/2" high (8" high including PROM cartridge).

Power Required: 120/240V ac, 60/50 Hz, <30 watts

Weight: 9 lbs.

Environmental Range

Temperature: 0 to +40° C, operating; 0 to +65° C, non-operating.

Humidity: 10% to 95%

OPTIONS AND ACCESSORIES

Option 001 Prom Reader/Programmer (includes 1 ea 073630 PROM cartridge)

P/N 073630 Prom Cartridge

Prom: UV Erasable

Capacity: 2K Bytes

P/N 073663 IEEE488 Bus Control Line Status Indicator with bus monitoring points.

All data subject to change without notice.

1.3 RECEIVING YOUR BUSSER

As soon as you receive your BUSSER, check the carton for evidence of damage or rough handling. If damage is found, or is suspected, notify the carrier and your Systron-Donner representative. Open the shipping container only in their presence. DO NOT USE THE INSTRUMENT UNTIL INSTRUCTED TO DO SO BY THE REPRESENTATIVE.

Use care in removing your BUSSER from the shipping container. Check immediately for loose or broken keys, bent or broken connectors, and dents or scratches on the panel surfaces. If damage of any nature is found, refer to the warranty instructions in the front of this instruction manual.

1.4 POWER UP!

Let's get started using the BUSSER by connecting power, and getting it to the 'ready' state. Plug the line cord attached to the rear panel of the BUSSER into a standard 120 VAC, 60 Hz outlet. Move the "ON-OFF" toggle switch on the rear panel to the up, or "ON", position. The front panel display may flash on and off momentarily. Observe the "READY" indicator on the left of the display illuminate. The display will indicate the version number of the software installed in your BUSSER.

Your BUSSER is now ready for operation.

1.5 EXPLORING THE KEYBOARD

1.5.1 Typewriter Keys

Before you learn to program the BUSSER, and to use it as an IEEE controller, let's explore the keyboard. The lowest four lines of the keyboard are laid out just like a standard typewriter keyboard. (See Figure 1-1) There are, however, some differences. On a standard keyboard, when you depress a letter without using the shift key, you get a lower case letter. The BUSSER has no lower case letters, so if you depress a letter without using the shift key you get an upper case letter.

1.5.2 Using the Shift Key

Depressing the shift key with the keys that have upper and lower markings, displays the punctuation characters that appear above the letters or numbers. We will use these punctuation characters later in programming.

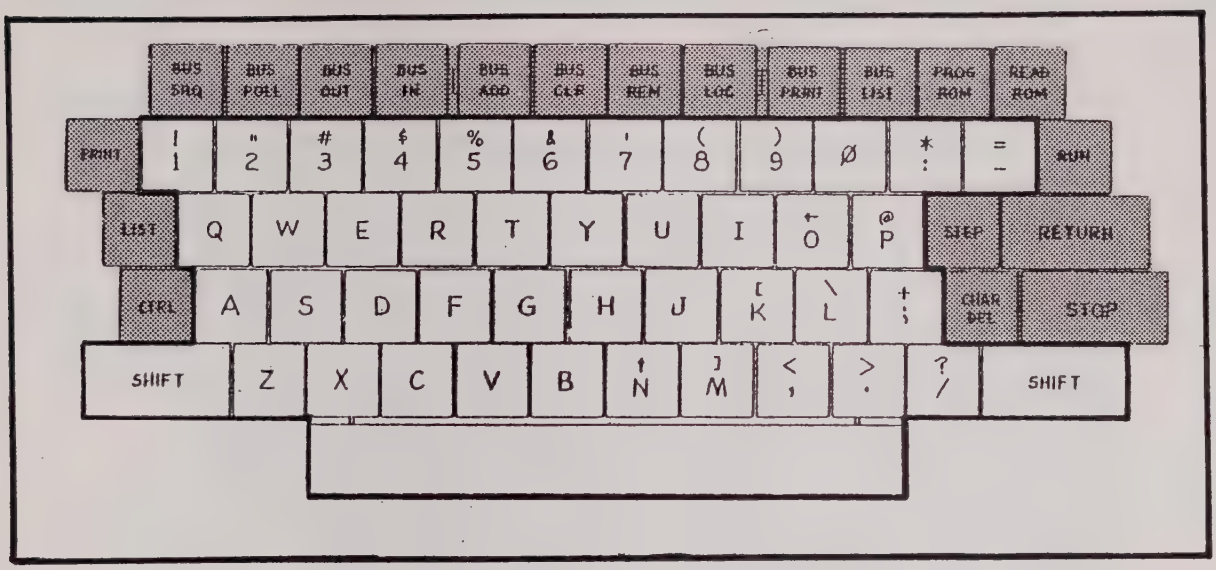


FIGURE 1-1. TELETYPE KEYBOARD

Try typing in a few characters to get a feel for the keyboard. First type the letters *H E L L O*.

The display should look like this:

```
HELLO
```

Depress STOP

Now try typing the letters *L L O* while using the shift key. The display should look like this:

```
\\←
```

The rest of the keys that show a symbol as well as an alphanumeric character will print one or the other depending upon whether the shift key is depressed.

1.5.3 Using the Character Delete Key

Let's suppose you made a mistake when you typed in the characters "HELLO". Try typing in the characters *H E L L O B U S T E R*. Suppose you wanted to correct BUSTER to BUSSER. Mistakes are easily corrected by using the CHAR DEL key. Depress the CHAR DEL key. Did you see the R disappear from the screen? Depress CHAR DEL two more times to erase the E and the T. Now you may retype the letters *S E R* to correctly spell the word BUSSER. Each time you depress the CHAR DEL key, the character

farthest to the right is erased. Using this key makes correcting mistakes typed onto the BUSSER screen very easy.

Let's explore some more of the keyboard.

1.5.4 Program Control Keys

Outside the alphanumeric keyboard is a set of keys called the program control keys.

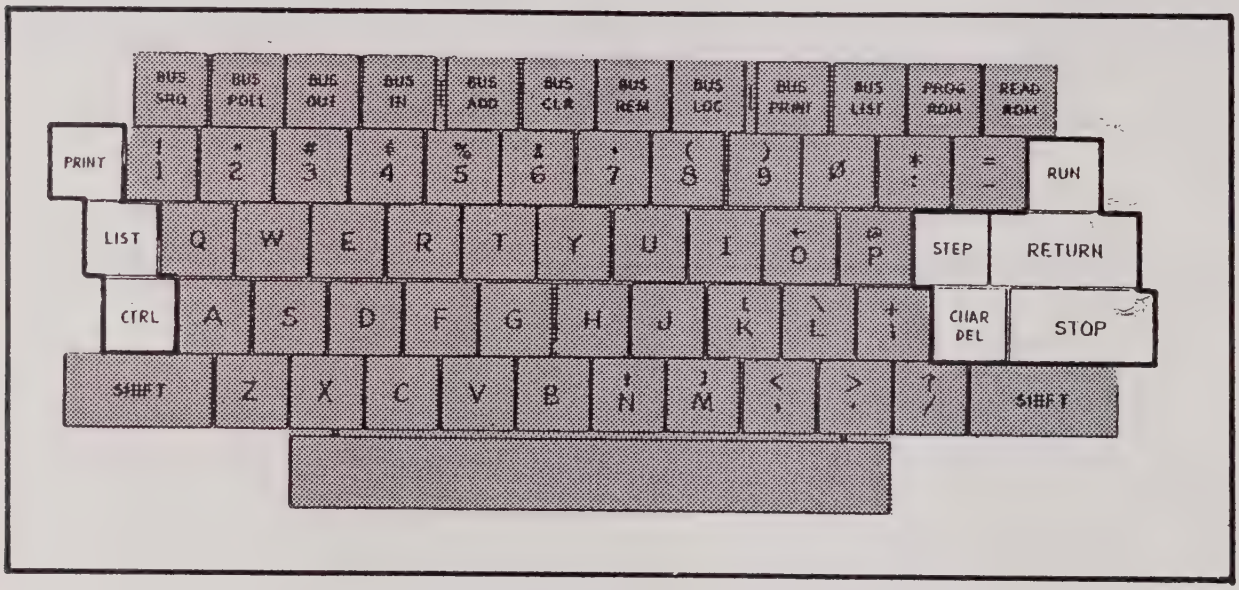


FIGURE 1-2. PROGRAM CONTROL KEYS

These keys are used in writing and executing programs. We will examine them further later on in Section I. (See Figure 1-2)

1.5.5 BUS Control Keys

The row of keys on the top of the keyboard, excluding the two keys on the far right, are BUS Control Keys (See Figure 1-3).

These powerful keys allow you to send complex commands on the BUS with a single keystroke. They are exclusive to the 3520 BUSSER. Their use will be elaborated later.

1.5.6 ROM Control Keys

The two remaining keys on the far right of the top row are ROM control keys. They are used to save programs on the PROM, or to recover programs already recorded in the PROM. Their use will also be elaborated later in the manual.

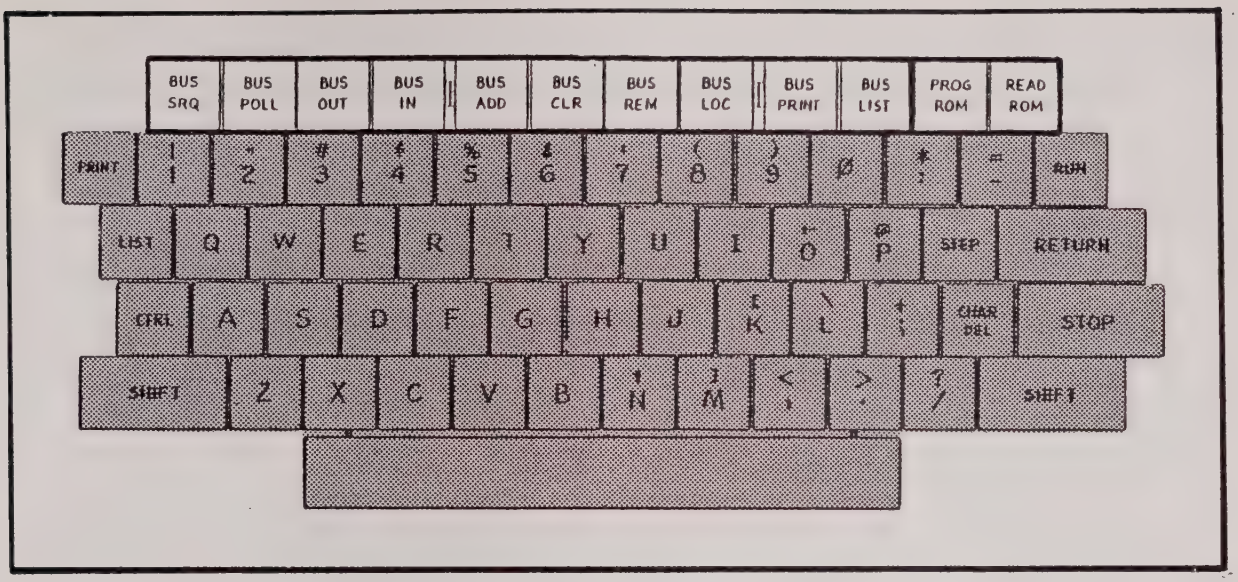


FIGURE 1-3. BUS AND ROM CONTROL KEYS

1.6 PROGRAMMING THE BUSSER

Now that you have explored the different types of keys on the keyboard, wouldn't you like to use them to write a program? The following section will describe the functions of the keys in more depth and show you how to use them to create a program.

1.6.1 Using the RETURN Key

Try typing in the following line:

HI *BUSSER* RETURN

Did you notice that the line stayed on the display until the RETURN key was depressed? EVERYTIME YOU TYPE A LINE, YOU MUST FOLLOW IT WITH THE RETURN KEY. The RETURN key tells the BUSSER that you are through typing a line and to process that line. This allows you to edit a line and get it typed correctly before the BUSSER processes the line.

1.6.2 Error Messages

After you have finished typing in the above letters, and depressed RETURN , the screen should show the following:

```
ERROR #1
```

This 'error' message just means that the BUSSER did not understand what you said to it. The BUSSER operates in a language called BASIC which was developed by programmers at Dartmouth University especially for making the resources of a computer quickly and easily available to those with no previous experience. If you speak to your BUSSER in words that are not in the BASIC language, you will receive an 'error' message. Look up this 'error' message in the ERROR dictionary on page 2-20. "Unrecognizable statement" just means you have not spoken BASIC properly. The following section will introduce the rules necessary to speak BASIC so that the BUSSER will understand you.

1.6.3 Using PRINT with Strings

Let's learn how to print statements on the screen without getting error messages. This time type the following:

PRINT *"HELLO BUSSER"* RETURN

The display should look like this:

```
PRINT "HELLO BUSSER"HELLO BUSSER
```

If you got an error message instead, look it up and then try again. The PRINT command tells the BUSSER to display whatever follows on the line. If you want to print groups of characters, called 'strings', such as "HELLO BUSSER", the strings must be surrounded by quote marks. The quotes tell the BUSSER to print out the message exactly as it appears within the quotes without any further processing.

1.6.4 Using PRINT with Variables

If you type in letters following the PRINT statement that are not in quotes, the BUSSER will interpret them as variables. Try using the PRINT statement by typing in the following:

PRINT *A* RETURN

The display should look like the following:

```
PRINT A Ø
```

With this statement the BUSSER will search its memory for the value of variable A and print that value. In the BUSSER's memory all variables are initially set to Ø. Because A has not yet been assigned a value, typing PRINT A results in a Ø on the display.
Let's use what we have learned about the PRINT statement to write a small program.

1.6.5 Programming

Up until now you have been typing in single lines containing instructions for the BUSSER. Suppose you wanted to collect a sequence of instructions for the BUSSER to perform as a group. This would be called a program. A program is nothing more than a sequence of instructions that is stored in the BUSSER's memory.

1.6.6 Immediate and Deferred Modes

The BUSSER will accept commands in two different 'modes': immediate mode and deferred mode. So far in your instructions to the BUSSER, you have been using the immediate mode. If you enter a command (such as PRINT "HELLO BUSSER") from the keyboard and follow by depressing the RETURN key, your instruction will be executed immediately. What distinguishes these immediate mode statements from statements in the deferred mode is that they have no line numbers. Statements in the deferred mode all have line numbers.

When a line number precedes an instruction to the BUSSER, its execution is deferred until later. The deferred mode is used to store statements in the BUSSER's memory so that you can execute them all at once as a program. When a RUN command is issued later the stored statements will execute in numerical order, starting with the lowest line #. In the following we shall see how deferred mode statements (those with line numbers) can be assembled to form a program.

1.6.7 Creating a 'GAS BILL' Program

Let's use the things we've learned about making statements on the BUSSER to create a program. With just a few easy instructions we can make a novel program that shows an ever-increasing 'gas bill'. Our 'Gas Bill' program combines the elements of BASIC you have learned to show an amusing, 'typical gas bill' of the 80's.

The first line of our program should contain a variable that will be the amount of the gas bill, and a label that identifies the variable as the amount of the gas bill. This first line of the program must also have a line number so the BUSSER will know

to execute this line first. Try typing the following line:

10 PRINT "GAS BILL SPACE = SPACE $";N;".00" RETURN

If you received an error message, look it up in the error dictionary on page 2-20 and retype the line accordingly. The first element on this program line is the line number. It is good programming practice to number lines in increments of 10 to 20 when first writing a program to allow for insertion of forgotten or additional lines that you might want to add before the program is complete. This is why we started with line number 10 and will follow with a line number 20, etc.

The next part of the line is just like a line we learned eariler with "HELLO BUSSER". The PRINT statement is followed by a phrase in quotation marks "GAS BILL = $". As you remember a PRINT statement followed by characters in quotes tells the BUSSER to print those characters on the screen. Unlike our earlier effort, however, the BUSSER will not PRINT the letters as soon as we depress the return key. This time the letters are part of program line which will not be printed until we run the program later (deferred mode).

1.6.8 Using the Semicolon

The next part of the line after the quotes is a semicolon. The semicolon serves to separate the different items you are telling the BUSSER to print. If you did not type in the semicolon, did you receive a 'Syntax Error' message? In BASIC you must always separate the items after a PRINT statement with a semicolon or a comma (a comma will print each item in column format).

1.6.9 Variables

After the semicolon you typed the letter N. This letter is a variable, or an item of data whose value can be changed by the program. Any letters from A to Z may be used as variables. Because N, a variable, follows a print statement, your BUSSER will print the value that is assigned to N. As you remember from our discussion of variables on page 1-9, all variables are initially assigned a value or '0'. The last element on the first line of your program is another string in quotes that the BUSSER will print on the screen. This string just prints a 'cents' indicator for the Gas Bill dollars.

1.6.10 Assignment

Now type second line of the program as shown:

20 SPACE *LET* SPACE *N=N+1* RETURN

This line assigns a value to the variable N. Each time this line is executed 'one' will be added to the old value of N, and this sum will be assigned to the new value of N. The net effect is to increase the value of N by one each time the line is executed. As you remember from our earlier discussion of variables (page 1-9), all variables are initially assigned a value of zero. N had the value of zero in the first line, so now the value is incremented to one. This line will give our gas bill a typical inflationary spurt every time it is executed.

1.6.11 Using GOTO

Now let's type the third and last line of our program. Try typing this line:

30 SPACE *GOTO* SPACE *10* * RETURN

GOTO is a branch statement. It commands the BUSSER to branch or 'GO TO' the line number specified after the GOTO statement. This statement is very useful because it can tell the BUSSER to execute any other step in program memory. In this program the GOTO statement tells your BUSSER to return to and execute the first line of the program, line 10.

Are you ready to run your first program?

If you typed in the lines correctly, you should have a line that prints a 'GAS BILL' label and an N variable (line #10), a line to increment (or increase), the N variable (line#20), and a line that branches back to line 10 (line #30).

> *NOTE: The BUSSER doesn't care about spaces in a program line. Spaces between words in BASIC are not evaluated and have no effect on the program. The use of the space bar above is merely to make the program lines more readable.

1.6.12 Using LIST

If you would like to check the program once before running it, you can use the LIST command to review the program. List shows each line of the program in sequence, from the smallest to the largest line number.
Try typing in:

You should see the following:

```
10 PRINT "GAS BILL = $";N;".00"
```

(new display)

```
20 LET N=N+1
```

(new display)

```
30 GOTO 10
```

(display clear)

1.6.13 Using STEP

Did these lines go by too fast? You can review each line at your own pace by using the STEP mode. Depress the STEP key once. You should see the STEP indicator above the 'READY' light illuminate. When you are in STEP mode each program line will stay on the display until the RETURN key is depressed. (You may notice that actually any key will serve this function) When you depress the RETURN key the next line of the program will be displayed until you depress RETURN again. (See pg 2-9 for a thorough discussion of STEP) LISTing the program in STEP mode allows you to check each line for errors before running the program.

1.6.14 Correcting Entire Lines

However, once a program line has been set up in the BUSSER's memory, as you have done here, you cannot change the line by using the CHAR DEL key. If you want to make corrections on a line, type the line number followed by the new line of text.

Your BUSSER will erase the old line and enter the new line in its place. Before proceeding, depress STEP once to go out of the step mode. The STEP indicator should be extinguished.

1.6.15 RUNning the Program

Are you satisfied that each line of your program is correct? If so, let's run the program. To do this just press RUN . If you wrote your program correctly, the display should show the following:

```
GASBILL = $0.00
```

The dollar amount should increase by a dollar every two seconds. Does that look like your gas bill?

1.6.16 Using STOP in Immediate Mode

Watching this program run too long might make you a bit uneasy! Would you like to learn how to 'stop' the program? The STOP command is designed just for this purpose. There are two different ways the STOP command may be used: in immediate and deferred mode. To use the STOP command in immediate mode, depress STOP now, while the program is running. Program execution will stop and the line that the program stopped at will be displayed as follows:

```
STOP AT LINE #10
```

Your display should look like the above, though the line number may be different. Suppose you wanted to restart the program. Try depressing RETURN . Did the display clear and the ready indicator illuminate? You can see that the program will not proceed from stop by pressing the RETURN key. Depress RUN to start the program once again. Now depress STOP to again stop the program. To restart the program again depress RUN . Did the program start again from the beginning? Whenever you have stopped a program, depressing RUN will restart the program from the beginning. If you wanted to restart the program where you interrupted its execution you can use the GOTO command. Type in:

GOTO 10 RETURN

Run clears the variables to zero, and the GOTO command starts the program while maintaining values of the variables.

The GOTO command when followed by an appropriate line number will restart the program where execution was stopped.

1.6.17 Using STOP in Deferred Mode

The STOP command may also be used in the deferred mode. Stop the program that is currently running by depressing STOP, so that you may type in a new program line. Type the following:

25 IF N = 150 THEN STOP RETURN

This new program line checks the value of N everytime line #25 is encountered in the program. Just as the line says, if N= (is equal to) 150 then stop (the program). This line will be executed each time the program increases the variable, and then the program will stop when the 'gas bill' reaches $150.00 The display should look like the following:

```
STOP AT LINE #25
```

The program may be restarted by depressing RUN.

1.6.18 Amending the Program

Does your actual 'gas bill' go up even faster than is shown in this program? If it does, you can amend the program by adding a line that will speed up the rate of increase. This is good practice in case you ever want to add a line to an existing program. First stop the program by depressing the STOP key. When you have stopped your program (or if it was already) try typing in the following line:

1 PAUSE 1

The pause statement controls the length of time the display is visible. The number following the PAUSE command is the length of time the display is visible in tenths of a second. Previously, each time the print command in line 10 was encountered, the display would be shown for 2 seconds. Now depress RUN. The important thing to note here is that any program can be amended simply by adding a line that does not already exist in the program. The BUSSER will automatically amend the program by placing the line in the appropriate place — in sequence with the other existing lines.

1.6.19 Recording the Program

NOTE: The following section only applies if a PROM cartridge is plugged into the top of the BUSSER. (P/N 73630)

Now that we've used our 'gas bill' program for a bit, maybe you'd like to set it aside to show someone later. To record the program for later use, depress the PROG ROM key. The display will look like the following:

```
SAVE TEXT IN CARTRIDGE [Y,N]?
```

If you answer by depressing Y, the program will be saved. If you answer by depressing N the BUSSER will return to ready mode. In our present case, depress Y, and follow by depressing RETURN. The display will clear for a few seconds and the light on top of the EPROM cartridge will illuminate during the program recording. When recording is finished, the display will look like the following:

```
SAVE TEXT IN CARTRIDGE [Y,N]? Y 1
```

The numeral one following the question mark indicates that the program saved is the first program in the EPROM.

1.6.20 Loading the Program

Now that you've saved the program, you might want to know how to recall the saved program. Depress the READ ROM key; the display should look like the following:

```
PROGRAM NUMBER?
```

Now type in the number of the program you would like to recall. Your 'gas bill' program should be the first program on the EPROM. If this is the case, type in: 1 RETURN.

The program has now been entered into the BUSSER's memory from the EPROM storage. You may run the program just as if you had just typed it in.

1.6.21 The PROM Cartridge

The cartridge that your program is now stored in will be referred to as the EPROM or PROM cartridge. The PROM cartridge stores programs by number, starting with program #1. The programs are then stored sequentially in a manner similar to a magnetic tape system. The PROM cartridge is supplied completely erased and ready for the entry of programs. The first program is placed at the beginning of the ROM, and subsequent programs are placed after the program entered most recently. When insufficient space is available for a program you wish to load, an error message will occur at this point without using the remaining space. A smaller program may be stored if desired.

When entering a given program into the PROM cartridge is complete, place the number and description of the program on the PROM label as soon as possible.

As good programming practice, place a remark statement at line #1 of every program, indicating the program name and function. This allows the programs to be read sequentially, and their purpose easily determined by LISTing the first line.

> CAUTION: Do not remove the PROM cartridge when the red indicator is illuminated. Permanent damage to the PROM may occur.

1.6.22 Self Test

The BUSSER features a self-test at power application (or when the verb 'START' is executed). If the self-test is sucessful, the version number of the software (the BASIC language) will be displayed. This version number allows you to be informed about the status of the software in the event of any factory change.

An unsucessful self-test will display the error message that follows:

```
XXXX YY ERROR
```

This error message will not, of course, be displayed if the failure is in the area required to operate the display.

In the above error, XXXX is the address of the first failure and YY is the exclusive 'OR' of the data stored in the memory cell compared to the data read from same. The ROM is also read and the number of bits compared to a check sum in the ROM. The address of the hardware is as follows:

0000 TO 007F	6802 Processor internal ROM error
0400 TO 0BFF	RAM memory area (2K bytes)
FFFF	Address of check sum for ROM

If any of these errors occur, suitable service should be done on the unit. The controller is still operational, but the memory errors will cause program text storage errors, variable storage errors, or errors in the execution of the BASIC language. The unit should be placed out of service as soon as possible.

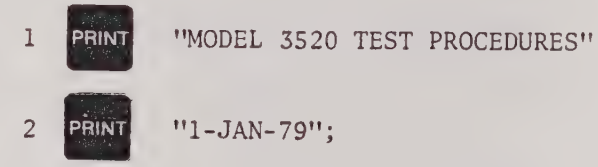
1.6.23 Autostart

The first program in the PROM cartridge will be self-starting, that is the program will load and automatically execute upon the application of ac power to the BUSSER. The self-starting feature can be used to identify a PROM cartridge, or used to allow operation in an unattended environment. A power failure will cause the solid state random access memory to 'forget', but the autostart feature can reload and restart the program.

The first program will be self-starting unless it is altered to be non-self starting by using the 'AUTOSTART:=" verb. 'AUTOSTART:=2' will cause program number 1 to be skipped over in the AUTOSTART search at power-up. If 'AUTOSTART:=3' is entered, now program number three will autostart. This is accomplished by altering the data in programs #1 and #2 to cause them to be skipped over in the autostart search at power up. The programs prior to the current autostart program cannot be made autostart, i.e. if AUTOSTART:=3 has already been implemented.

1.6.24 PROM Control using AUTOSTART

The AUTOSTART feature can be used to generate a short program at the beginning of the PROM space as follows:



If the PROM is in place when the power is turned on, this program will display the PROM's identification. The semicolon on the bottom line prevents the clearing of the display when the BUSSER returns to 'ready'.

This program allows the PROM to be identified and prevents any application program from starting. If you desire not to self-start the PROM, execute 'AUTOSTART:=22', or some other number greater than the number of programs in the ROM. This will supress the self-start procedure until a new program is stored. The new program will be self-starting unless the AUTOSTART verb is used to alter it.

1.7 PROGRAMMING WITH IEEE INSTRUMENTS

1.7.1 IEEE Programming

Now that you know how to program the BUSSER, let's try some programming that can be used with IEEE-488 equipment. The following section on programming uses the Model 3520 as an IEEE-Bus controller, instead of just as a desk top computer as in the previous 'Gas Bill' program.

The IEEE device that is used in the following program is a Systron-Donner Model M107 Precision D.C. Voltage Source. However, do not be concerned if an M107 is not available; this program will run the same, even if an M107 is not connected.

Immediately following are procedures for connecting the M107 to the BUSSER. Most IEEE compatible devices will follow a similar procedure.

1.7.2 Connecting the IEEE Instrument

Let's connect the M107 to the IEEE Bus:

1. Follow the power - up procedures for the M107 as shown below.

    a. Set the line switch to the voltage that will be used, 115V or 230V.

    b. Verify that the correct fuse is installed.

    c. Connect the power cord to the rear panel of the instrument and to the appropriate outlet.

2. Connect an IEEE cable between the BUSSER and the M107.

3. Check the IEEE address of the M107 as indicated by the lights on the front panel. If the address is identical to any other instrument you are using in your present BUS system, change to a new address via the switches on the rear panel.

When the M107 (if available) is connected, you are ready to begin entering the following program. Remember, it is not necessary to have an M107 to run the following program. Type NEW to clear the BUSSER memory for a new program.

1.7.3 Using BUSCLEAR

The following program will increment the voltage output of the M107 from 0 to +9.99000V in 10 mV increments.

Type in the following:

10 BUS CLR RETURN

The display should look like this:

```
10BUSCLEAR
```

Note that it is unnecessary to insert a space between 10 and the BUS CLR command.

Try typing in the following:

LIST *10* RETURN

The display should look like this:

```
10 BUSCLEAR
```

The BUSSER always inserts spaces after line numbers, so you didn't have to insert a space when you typed in the original line.

You might ask, 'what does the BUSCLEAR command do in our program?' The BUSCLEAR command should appear at the start of every program. It places each unit on the BUS in a known quiescent (inactive) state. It does this by strobing the IFC, or interface clear line, which stops all communication on the BUS.

1.7.4 Using BUSREM

Try typing in the following line:

20 [BUS REM] [RETURN]

The display should look like this:

```
20BUSREMOTE
```

The BUSREMOTE command should always be the second command in a program after BUSCLEAR. The BUSREMOTE command sets up the initial conditions for operating instruments under remote control. This is accomplished by pulling the REN (Remote ENable) line true. (Not all instruments monitor the REN line).

1.7.5 Using BUSADDRESS

Try typing in the next line of the program:

30 [BUS ADD] *10101* [RETURN]

The display should look like this:

```
30BUSADDRESS 10101
```

As you recall, if the line is LISTed at a later time, there will be a space between '30' and BUSADDRESS, so you don't need to type one in now.

This command tells the BUSSER that a certain instrument (specified by the address given after the command) will be used in all future BUS transactions. This command stores this address in the BUSSER and is used for all communication, i.e. talking and listening, until a new address is specified. A second address may be used as follows which specifies an additional listen-only device such as a printer:

[BUS ADD] *10101,00111*

Where 00111 is the address of the printer.

If the M107 address in the example shown above is not the same as on your M107 address (which it probably isn't) set the switches on the rear panel of the M107 from the existing address to the new address, 10101. If the unit has switches for the address, the number represents the switch position with switch #1 on the right.

1.7.6 Using BUSOUT

Try typing in the following line:

40 [BUS OUT] *D0E0F0R1P1S* [RETURN]

The display should look like this

```
40BUSOUT DØEØFØR1P1S
```

BUSOUT 'talks' the IEEE command string onto the BUS for use by the addressed instrument. The letters used in this case are control letters for the M107 as shown in the following table.

TABLE 1-2. VOLTAGE ENTRY CHARACTERS

ASCII Character	Data Function
A	Stores following numerical character in 10^5 (MSD) register.
B	Stores following numerical character in 10^4 register.
C	Stores following numerical character in 10^3 register.
D	Stores following numerical character in 10^2 register.
E	Stores following numerical character in 10^1 register.
F	Stores following numerical character in 10^0 (LSD) register.

D followed by a 0 tells the M107 to make the number in the 10^2 place voltage digit a zero. E followed by a 0 tells the M107 to make the 10^1 voltage digit a zero, and similarly for the letter F and the 10^0 voltage digit. The following control code is used for range:

TABLE 1-3. RANGE ENTRY CHARACTERS

ASCII Character	Data Range Register
Enter "R" followed by:	
0	1 volt range
1	10 volt range
2	100 volt range
3	1000 volt range

When R is followed by a 1 in our BUSOUT line it tells the M107 that it should go the 10 volt range. Polarity works similarly.

TABLE 1-4. POLARITY ENTRY CHARACTERS

ASCII Character	Data Polarity Register
Enter "P" followed by:	
0	Standby (output terminals shorted)
1	"+" Positive Polarity
2	"-" Negative Polarity

The control letter 'P' followed by a 1 tells the M107 to go to positive polarity.

These control letters by themselves will not cause the M107 to assume the desired voltage, range and polarity. The strobe character, 'S' must follow any desired control characters. The strobe character switches data received into the M107's working registers. The output voltage directly follows the M107's working registers.

Let's review the characters we have 'talked' on the Bus to the M107.

TABLE 1-5. IEEE - 488 PROGRAMMING FORMAT MEANING

Program Bits							ASCII Character	Function
7	6	5	4	3	2	1		
1	0	0	0	0	0	1	A	10^5 Digit
1	0	0	0	0	1	0	B	10^4 Digit
1	0	0	0	0	1	1	C	10^3 Digit
1	0	0	0	1	0	0	D	10^2 Digit
1	0	0	0	1	0	1	E	10^1 Digit
1	0	0	0	1	1	0	F	10^0 Digit
1	0	1	0	0	0	0	P	Polarity
1	0	1	0	0	1	0	R	Range
1	0	1	0	0	1	1	S	Strobe

D0E0F0 sets the 10^2, 10^1 and 10^0 voltage registers at zero. R1 sets the range at 10V. P1 sets the polarity at '+' and, the letter S strobes the M107, commanding the above data into the working registers. You might ask 'What about the 10^5, 10^4 and 10^3 voltage digits?' We'll explore that in detail as we look at the next line of the program.

1.7.7 Using FOR - NEXT loops

Try typing in the following lines.

50 FOR A = 0 TO 9 RETURN

60 FOR B = 0 TO 9 RETURN

70 FOR C = 0 TO 9 RETURN

To check your program lines, depress the following sequence of keys STEP , LIST 50,70 RETURN . After you verify each line depress RETURN to check the next line.

The display should appear as follows:

```
50 FOR A = 0 TO 9
```

(press RETURN for new display)

```
60 FOR B = 0 TO 9
```

(press RETURN for new display)

```
70 FOR C = 0 TO 9
```

(press RETURN clears display, press STEP to leave step mode)

The commands you have just entered in the program are the beginning statements in the programming technique called 'loops'. Loops use the FOR and NEXT statements to control the number of times a section of a program is executed repeatedly. The following example illustrates a simple FOR-NEXT loop. The statements in the loop are executed three times before the program continues with normal execution.

```
210 FOR I = 1 TO 3
220
 .      [statements to be executed repeatedly]
 .
 .
260
270 NEXT I
 .
 .
```

Line 210 in this example is the first statement in the loop. This statement controls the number of times the loop is executed. In line 210, the variable I is specified as the index and is assigned a starting value of 1. The ending value of I is specified as 3.

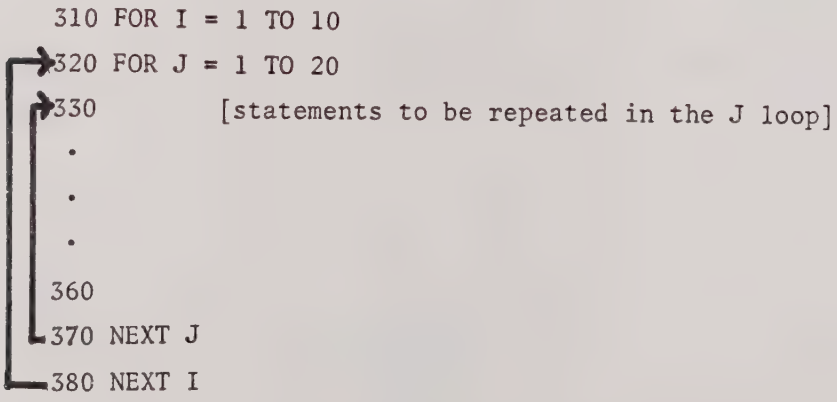
After the FOR statement is evaluated in line 210, lines 220 through 260 are executed in sequence. During this time the assigned value of I remains at 1. When line 270 (the NEXT statement) is executed, +1 is added to the value of I and

the new value (2) is compared to the ending value (3) as specified in line 210. In this case, the new value of I (2) is not greater than the specified ending value (3), so the loop is entered again and lines 220 through 260 are re-executed. As these statements are executed, the assigned value of I remains at 2.

At the end of the second pass through the loop, line 270 is re-executed. Again the increment +1 is added to the current value of I and the new I is compared to the ending value specified in line 210. This time the new value (3) is equal to the specified ending value, but not greater than the specified ending value; therefore the statements in the loop are executed a third time. During this pass, the assigned value of I remains at 3.

At the end of the third pass, the increment +1 is added to the current value as before, but this time the new value (4) is found to be greater than the specified ending value (3). When this occurs the loop is exited, the lines following line 270 are executed, and the program continues executing in sequence.

Our loop is actually three loops in what is called a "nested loop" format. Nested loops are illustrated in the following example.



In this example, lines 330 through 360 are executed 20 times for each pass through the I loop. After the program exits from the I loop, the statements inside the J loop will have executed 200 times.

Your program lines 50,60, and 70 are in a nested loop similar to the above example. After you type lines 80 through 90 (which will be shown in the next two

sections, 1.7.8 and 1.7.9 of this manual), you will type in the lines containing:

```
NEXT C
NEXT B
NEXT A
```

(Do not type in these lines at this time)

The resulting set of loops will execute lines 80 through 90, inside the innermost loop (the 'C' loop), ten times for each pass through the B loop. The 'B' loop will in turn execute ten times for each pass through the A loop. For each pass through the A loop, the C loop will execute 100 times.

This format of nested loops allows the variables A, B and C to go from zero to 999 in a decimal manner in which A is the hundreds digit, B is the tens digit and C is the ones digit.

1.7.8 Using BUSPRINT

Try typing in the following line

80 [BUS PRINT] *"A";A;"B";B;"C";C;"S"* [RETURN]

The display should show the following:

```
80BUSPRINT "A";A;"B";B;"C";C;"S"
```

The BUSPRINT statement works just like the PRINT statement (See page 1-8) but talks the data on the BUS to the addressed unit. As with the PRINT statement the characters in quotes are interpreted as letters, and the characters without quotes are interpreted as variables. This program line talks the control letters of the M107 on the BUS. The M107 receives the letter A followed by a numeral (the value of variable A). The M107 interprets this as a command to change the 10^6 voltage digit to the numeral following "A", which is the value of the variable A. Similarly, when the M107 encounters a "B", the 10^5 voltage digit is assigned the value of the variable B. The 10^4 voltage digit works in a manner similar to the 10^5 voltage digit. The last character on the line, "S", is the Strobe character for the M107. When the M107 receives an "S" character on the BUS, voltage digits that have been entered go from the interim to the working registers and are shown on the output.

As you recall from our discussion of loops, the letters A, B, and C increment upwards from 0 to 999. The BUSPRINT command, inside this loop, increments the voltage of the 10^6, 10^5, and 10^4 digits from 0 to 999. The "A""B" and "C" letters tell the M107 which digit to program and the A,B, and C variables tell the M107 what values the digits should have, from 0 to 999.

1.7.9 Using PRINT in an IEEE Program

Try typing in the following line:

90 PRINT *"A";A;" B";B;" C";C* RETURN

The display should appear as follows before the RETURN key is depressed.

```
90PRINT "A";A;" B";B;" C";C
```

The PRINT command is just like the PRINT command in the GAS BILL program. It is a command to print whatever follows on the line. The print command will display the A,B, and C letters and follow each with the A,B and C variable values. The result is the displaying of exactly what is being talked on the BUS to the M107. The spaces before the B and C letters aid in making the display easier to read.

1.7.10 Using NEXT

Try typing in the following lines:

100NEXTC RETURN

110NEXTB RETURN

120NEXTA RETURN

To display the finished lines depress LIST 100,120. The lines should look like the following:

```
100 NEXT C
```

```
110 NEXT B
```

```
120 NEXT A
```

These NEXT statements are the last statements in the FOR — NEXT loops we created in section 1.7.7. That is, line 100 closes the loop created by line 70; line 110 closes the loop created by line 60; and line 120 closes the loop created by line 50. When the variables A,B and C reach the values 9,9, and 9, the program will have completed line 120 for the last time and the program will stop executing.

1.7.11 Listing the Program

Depress LIST , STEP , RETURN to check the program lines you have typed in so far. The lines should appear as follows:

```
10 BUSCLEAR
20 BUSREMOTE
30 BUSADDRESS 10101
40 BUSOUT DØEØFØR1P1S
50 FOR A=0 TO 9
60 FOR B=0 TO 9
70 FOR C=0 TO 9
80 BUSPRINT "A";A;"B";B;"C";C;"S"
90 PRINT "A";A;" B";B;" C";C
100 NEXT C
110 NEXT B
120 NEXT A
```

You are in step mode, so remember to depress RETURN to see each line.

1.7.12 Running the Program

Depress STEP RUN to leave STEP mode and run the program. Depressing STEP is not necessary to run a program, however LISTing the program in section 1.7.11 left you in STEP mode. Remember that in STEP mode you would have to depress RETURN to execute each line.

If you typed in the program correctly, the display should look like the following:

```
A 0 B 0 C 0
```

Where the numbers count up to 999. If an M107 is connected you will see the remote light illuminate, the front panel controls become inoperative, and the voltage increments upwards from +0.00000 to +9.99000. If you want to save this program follow the instructions in section 1.6.19 "Recording the Program". The program may be reloaded for use by following section 1.6.20.

1.7.13 AUTO-VERBS

The verbs used in the BASIC language may be generated with two-keystrokes instead of typing our the entire verb letter by letter. To generate the verbs in Table 1-6, depress and hold the CTRL key while depressing the appropriate letter or number.

TABLE 1-6 AUTO-VERBS

CONTROL	
A	ATN (Attention)
B	Recall
C	Remark (Comment)
D	Dim
E	Step
F	For
G	Get (group execute trigger)
H	Stop (halt)

TABLE 1-6 AUTO-VERBS (Cont'd)

CONTROL	
I	I$ (Input buffer)
J	DCL (Device clear)
K	SDC (Selected device clear)
L	LLO (Local Lockout)
M	Carriage return
N	Next
O	O$ (Output buffer)
P	Pause
Q	If
R	Return
S	Gosub
T	Goto
U	UNL (Untalk)
W	Wait
X	Then
Y	GTL (Go to local)
Z	Mem-Info
/	Input
0	New
1	Start
2	Cart-Info
3	Mem-Info
4	Autostart:=
5	Recall
6	Let

These single stroke keys aid in rapid program generation, but are not required to be used. The same text will be generated by entering the verbs as character strings.

CHAPTER 2

COMMAND DICTIONARY

2.1.1 PROGRAM CONTROL KEYS

Key	Display	Description
RUN		Starts execution of program in memory.
STOP	STOP AT LINE 40	Stops program execution; displays line number where execution was stopped.
STEP	(Step light illuminates)	Selects single step mode (See page 2-9).
LIST	LIST	Lists current program.
	LIST 100	Lists program line #100.
	LIST 100, 200	Lists program line numbers 100 through 200.
PRINT	PRINT 123	Prints the #123 on the display.
	PRINT A	Prints the value of program variable A.
	PRINT A;B;C;	Prints three program variables separated by spaces.
	PRINT "ABC"	Prints the string 'ABC'.
CHAR DEL		Deletes the last character entered into the display.

PROGRAM CONTROL KEYS (Cont'd)

Key	Display	Description
RETURN		Causes the execution of displayed command in immediate mode, or the entry of program lines into memory.
CTRL		Used in conjunction with keys to produce automatic generation of verbs. (See page 1-30).

2.1.2 BUS CONTROL KEYS

Key	Display	Description
BUS SRQ	SERVICE REQUEST	Displays the state of the 'SRQ' line of IEEE bus to see if any unit requires service.
	SERVICE REQUEST A	Assigns the state of the SRQ line to variable A (0-no service, 1-service) (See page 2-11).
BUS POLL	SERIAL POLL	POLLS addressed unit. Displays status byte indicating what type of service, if any, is required.
	SERIAL POLL A	Assigns status byte to variable A (0-255) (See page 2-11).
BUS OUT	BUSOUT ABC	Talks characters 'ABC' on Bus.
BUS IN	BUSIN	Displays talked data from addressed unit.
	BUSIN A	Assigns up to four numeric characters to A.
	BUSIN A,B	Assigns four low order digits in 'B', remaining digits (four or less) to A (See page 2-12).

BUS CONTROL KEYS (Cont'd)

Key	Display	Description
BUS ADD	BUSADDRESS 00001	Sets address for all bus Transactions to unit with address 00001.
	BUSADDRESS 00001,00010	Includes an additional address for a listener only device such as a line printer.
BUS CLR	BUSCLEAR	Clears the Bus by asserting 'IFC' (Interface Clear).
BUS REM	BUSREMOTE	Sets instruments on the Bus to remote mode by asserting 'REN'. (Remote Enable)
BUS LOC	BUSLOCAL	Sets units to local mode.
BUS PRINT	BUSPRINT "ABC"	Talks the character string 'ABC' with carriage return and line feed (CR/LF)
	BUSPRINT "A";A;"B";B	Talks the character A, value of variable A, character B, value of variable B, with CR/LF. (See page 2-13).
BUS LIST	BUSLIST	Talks listing of program in memory with CR/LF.
	BUSLIST 100	Talks line #100.
	BUSLIST 100, 200	Talks lines #100 through #200.

2.1.3 ROM CONTROL KEYS

Key	Display	Description
PROG ROM	SAVE PROGRAM IN CARTRIDGE [Y,N]?	Depressing Y will save the program in current memory in the PROM cartridge. N aborts to ready.
READ ROM	PROGRAM NUMBER?	Asks for the number of the program in the ROM cartridge that the user desires to load into the BUSSER. Number followed by RETURN deletes existing memory and enters specified ROM program.

2.1.4 HIGH LEVEL LANGUAGE

2.1.4.1 ASSIGNMENT

Command	Example	Description
LET	*LET A = 123* or *A = 123*	Assigns a value to a variable.
INPUT	*10 INPUT A*	Accepts value of variable A from keyboard.
	10 INPUT A,B	Accepts value of two variables from keyboard, if separated by any delimiter.
RND	*A = RND(100)*	Assigns a random number ranging from but not including zero to 100.

2.1.4.2 ARRAYS

Command	Example	Description
DIM	*10 DIM A (10,10)*	Specifies an 10 by 10 array or matrix.
	20 INPUT A(1,1)	Assigns value from keyboard to element A(1,1)
	30 LET A(2,2) = 123	Assigns value to element A(2,2)

2.1.4.3 PROGRAM BRANCHING

Command	Example	Description
GOTO	*10 GOTO 40*	Branches to a specified line (40), skipping over intervening lines.
	20 GOTO A	Branches to line # equal to the value of A.
GOSUB	*50 GOSUB 80*	Branches to a subroutine (at line 80).
RETURN	*90 RETURN*	Subroutine exit, returns to line after most recent GOSUB.

2.1.4.4 ITERATION

Command	Example	Description
FOR ... NEXT	*100 FOR I = 1 TO 10* • • • *190 NEXT I*	Loop control. Performs all instructions between FOR and NEXT as many times as indicated by the index variable. (See page 2-14 for example of iteration.)

2.1.5 ARITHMETIC OPERATORS

Symbol	Example	Description
+	*10 C = A+10*	Addition - Adds 10 to A and assigns value to C.
*	*20 C = A*10*	Multiplication - Multiplies A by 10 and assigns value to C.
/	*30 C = A/10*	Division - Divides A by 10 and assigns value to C.
-	*40 C = A-10*	Subtraction - Subtracts 10 from A and assigns value to C.

See example page 2-15 for order of operations.

2.1.6 NUMERICAL DECISIONS

Symbol	Example	Description
=	*10 IF A=B then print "GOOD TEST"*	A 'equals' B
<>	*20 IF A<>B then print "CHECK UNIT"*	A 'does not equal' B
<	*30 IF A<B then C = 123*	A 'is less than' B
>	*40 IF A>B then C = 456*	A 'is greater than' B
<=	*50 IF C<=D then E = 123*	C 'is less than or equal to' D
>=	*60 IF C>=D then E = 456*	C 'is greater than or equal to' D

See example page 2-28 for more numerical decisions.

2.1.7 SYMBOLS

Symbol	Example	Description
;	*1000 PRINT "TEST";* *1020 PRINT " THIS";* *1030 PRINT " UNIT";*	Supresses the carriage return/line feed which normally clears the screen after each print statement.
,	*1050 PRINT "A",A,"B",B*	Formats printed data by placing eight spaces between data values. Supresses CR/LF.
O$ I$	*2000 PRINT O$(1)*	Accesses input (I$) and output (O$) buffers. This example prints the ASCII value of the first character in the output buffer. (See page 2-17).
"	*2050 BUSPRINT "TEST"*	Character string enclosures.
( )	*2100 LET C=(A+B)*A*	Determines order of operations.

2.1.8 MISCELLANEOUS

Statement	Example	Description
MEM-INFO	*MEM-INFO*	Displays the number of bytes in memory used by a program and the number of bytes unused.
PAUSE	*10 PAUSE 20*	Determines time of display presentation. Number following statement specifies time in tenths of seconds (This example: 2 seconds).
WAIT	*20 WAIT 10*	Causes delay in program execution. Number following statement specifies time in tenths of seconds. Must be reasserted for each desired delay.
REM	*30 REM - THIS IS A TEST PROGRAM*	Allows non-executable text to be inserted for documentation of the program.
NEW	*NEW*	Deletes current program from memory, sets all variables to zero.
TAB	*40 PRINT TAB (N); A*	Displays value of A at N+1 position on screen. (See example page 2-19).
RECALL	*RECALL*	Displays contents of input Buffer.

MISCELLANEOUS (Cont'd)

Statement	Example	Description
RECALLO	*50 RECALLO*	Displays contents of BUSOUT buffer; may be used as part of a program.
ATN	*60 ATN 63*	Allows codes to be talked on the bus with ATTENTION asserted. This example talks the unlisten character '?'.
AUTOSTART:=	*AUTOSTART:=*	Declares which program will be self starting (See page 1-17).
CART-INFO	*CART-INFO*	Examines the program storage in the prom cartridge. Displays the number of bytes used, the number of programs, and which program is the autostart program.(See page 2-19.)
START	*START*	Executes the autostart program.
RND	*50 PRINT RND(100)* *A=RND(100)*	A function generating a random number up to but not including the numbers used as parameters (This example generates a number from 1 to 99).

2.1.9 IEEE CONTROL CODES

Code	Example	Description
LLO	*1000 LLO*	Local Lockout, reasserts address.
DCL	*1000 DCL*	Device clear.
GTL	*1000 GTL*	Go to local; reasserts address.
GET	*1000 GET*	Group execute trigger.
SDC	*1000 SDC*	Selected device clear, reasserts address.
UNL	*1000 UNL*	Unlisten.
UNT	*1000 UNT*	Untalk.

2.2 EXAMPLES

2.2.1 STEP Mode

STEP is a mode controlled by the STEP key. When the key is depressed the BUSSER enters STEP mode and the 'step' light illuminates. If the STEP is depressed again, the BUSSER leaves STEP mode and the 'step' light goes out. In STEP mode the following effects occur:

1. In the immediate mode using LIST with 'step' enabled, each program line remains on the display until the RETURN key is depressed.

2. In the run mode with 'step' enabled, each program line and IEEE bus statement is displayed but not executed until the RETURN key is depressed. In this mode the display has an arrow symbol before the text indicating step mode.

The step mode may be entered at any time by depressing the STEP key. If the program is executing, the program will stop at the line executing, and display the next line. Depressing the RETURN key will execute the displayed line, and cause the next line of the program to be displayed. If it is desired to run the program without stepping program lines, depress STEP to turn off 'step' mode and depress RETURN to go back to a normally running program.

The step light may illuminate for other reasons besides stepping through programs. Under certain conditions the step light will illuminate by itself. The step light will illuminate when the controller is waiting for a key to be depressed, such as when an input statement is executed, or when an error message is displayed.

The step mode light illuminated in this case indicates that the program is waiting for a key to be depressed to continue. It is not necessarily an indication of being in the step mode.

If step mode is enabled before or during the running of a program, IEEE bus statements will be displayed in the following format. (see Figure 2-1)

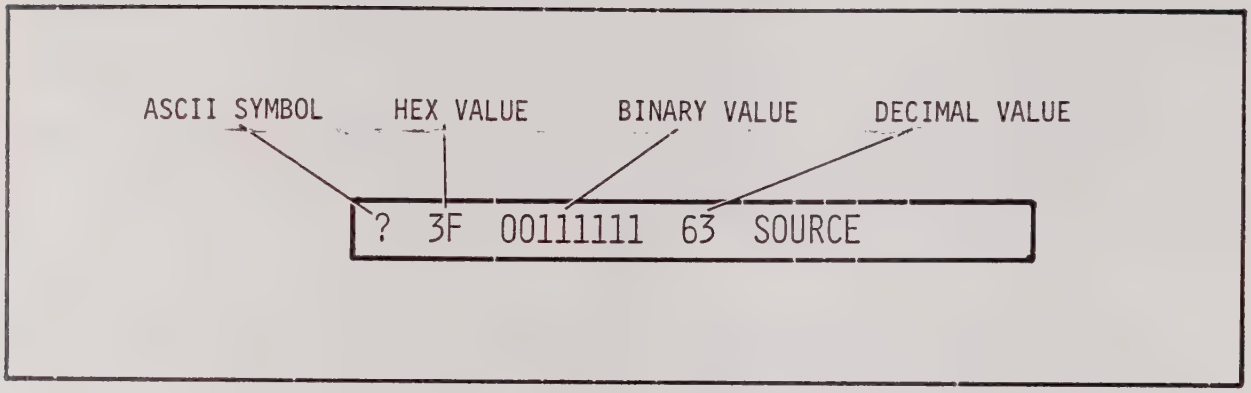


FIGURE 2-1. IEEE STATEMENTS

When 'SOURCE' is displayed, this IEEE statement shows what the software directs to appear on the BUS. (This can be verified with the P/N 073663 Control line status indicator). When 'ACCEPT' is displayed the IEEE statement being returned from the device, as in a BUSIN statement, will be displayed.

The first statements shown here will appear for the first occurance of the bus out statement, indicating the transmission of an 'unlisten' code. The display will continue until all of the IEEE characters have been transmitted. Non printing ASCII characters are displayed as an underscored square with the hex, binary, and decimal values displayed.

The step mode is also executable in a program to produce a 'break' for program debugging. The statement could be as follows:

1022 IF A>24 THEN STEP 1 — If the value of A exceeds 24 then go to the step mode.

Execution of this statement will display the next statement in the program. The step mode will continue until the STEP key is depressed or the statement 'STEPO' is executed, either in the program or from the keyboard. The value after the step statement may be a variable.

2.2.2 SERVICE REQUEST

The service request statement (generated by depressing [BUS SRQ]) returns the state of the SRQ line on the IEEE bus. If a variable is specified (such as [BUS SRQ] A) the value zero or one will be placed in the variable named. If no variable is specified, the value zero or one will be displayed.

The service request statement can be used in conjunction with a numerical 'IF' statement to determine the state of the 'SRQ' line as follows:

1000 [BUS OUT] *A1B2C3*	Program addressed instrument
1010 [BUS SRQ] *A*	Test service request
1020 IF A=1 THEN GOSUB 2000	Gosub to service routine

The service routine used in the GOSUB statement is elaborated in the following section.

2.2.3 SERIAL POLL

The serial poll statement (generated by depressing [BUS POLL]) polls the addressed unit for a byte of data (status byte) that indicates whether or not the unit needs service. If a variable is placed after the serial poll statement, the value of the status byte will be assigned to that variable. If no variable name is placed after the statement, the decimal value of the data in the status byte will be displayed.

If the addressed instrument is requesting service, the value will be equal to or greater than decimal sixty four (Bit 6 set). This can be determined easily by a numerical 'IF' statement. If the value is greater than 64, then the instrument is indicating the type of service required. The program would appear as follows for two instruments:

1010 [BUS ADD] *00001*	Set up address of first unit
1020 [BUS POLL] *A*	Serial Poll - assign to A

```
1030 IF A <64 THEN 1100                          Not this unit
1040 IF A-64=0 THEN PRINT "UNIT #2 NEEDS SERVICE"
1050 IF A-64=1 THEN PRINT "UNIT #2 IS OUT OF PAPER"
1100 BUS ADD 00010                               Set up for second unit
1110 BUS POLL A
1120 IF A <64 THEN RETURN                        Return from the subroutine
1130 PRINT "UNIT #3 NEEDS SERVICE"
1140 RETURN
```

The service routine (example program only) displays the instrument(s) that require service. The first unit returned two different service request conditions determined by a zero or one in the first bit. This was detected by two 'IF' statements with different messages.

2.2.4 BUSIN

The following elaborates the use of the BUSIN command (generated by depressing BUS IN). BUSIN displays data strings talked from the addressed unit. The sequence of operation is as follows:

1. The string received is searched to the first non-numeric delimiter that appears after the first number in the data.

2. The four least order digits are placed in the second variable, and the remaining digits (four or less) are placed in the first variable.

3. The string is then searched to the next limiter and the value placed into the third variable. (Plus or minus signs are not considered delimiters)

Examples:

INPUT STRING ON IEEE BUS	BUSIN STATEMENT	VALUES
VDC12345E-3	BUS IN *A,B,C*	*A=1 B=2345 C=-3*
123:12:16:54	BUS IN *A,B,C,D*	*A=123 B=12 C=16 D=54*
VDC-12345E-3	BUS IN *A,B,C*	*A=-1 B=-2345 C=-3*
OLVDC00000E-2	BUS IN *A,B,C*	*A=0 B=0 C=-2*
VDC-00001E-2	BUS IN *A,B,C*	*A=0 B=-1 C=-2*
12345678901@E0	BUS IN *A,B,C,D*	*A=123 B=4567 C=8901 D=0*
00000000123@E3	BUS IN *A,B,C,D*	*A=0 B=0 C=123 D=3*

2.2.5 BUSPRINT

The BUSPRINT Command (generated by depressing BUS PRINT), can be used to print characters on the GPIB bus similar to the BUSOUT command. Unlike BUSOUT, the BUSPRINT command can be used to print variables on the BUS. Any characters following the BUSPRINT command that are not surrounded by quotes will be interpreted as variables.

The following program shows this use of the BUSPRINT Command. This example iterates a Model M107 programmable voltage source.

```
1000 FOR A = 0TO9
1010 FOR B = 0TO9
1020 FOR C = 0TO9
1030 BUS PRINT "A"; A; "B"; B; "C"; C;
1040 BUS PRINT "DOEOFOR1P1S"          Send remainder of setup data
1050 NEXT C
1060 NEXT B
```

```
1070 NEXT A
1080 STOP
```

On the first pass through the loop, line 1030 will be interpreted as BUSPRINT A0B0C0; each successive pass through the loop will increment the numerical value of the variables A,B,&C. "A", "B" and "C" will remain intact as characters.

2.2.6 ITERATION

A variable can be iterated, that is, increased in value by one each time a loop is performed, by using the following statement.

`1000 FOR I = 1T010`	Sets I=1 initially
`1100 PRINT I`	Displays the current value of I
`1200 NEXT I`	Increments I - If I less than or equal to 10 then goto 1100

When I is equal to 10, the loop is performed for the last time, and the next statements in the program are executed.

The following program will enter values in an array in one iterative loop, and display the values in a second loop.

`100 DIM A (10)`	Space for ten variables
`1000 FOR I = 1 TO 10`	
`1100 INPUT A(I)`	Enters values of variables from keyboard
`1200 NEXT I`	
`2000 FOR I = 1 TO 10`	Sets up display of array
`2010 PRINT A(I)`	Displays values entered
`2020 NEXT I`	

An iterative loop may contain variables or expressions for the start and stop values, as in the following program:

```
1000 FOR I = A TO B*2                    Sets up I equal to variable A
1010 PRINT I
1020 NEXT I                              Increments I - branches to 1010 IF
                                         I <= 2 Times variable B
```

2.2.7 NUMERICAL OPERATORS

The following are some typical statements using the numerical operators:

```
2000 LET A = 10
2020 LET B = 25
2030 LET C = A + B                       Assigns 35 to variable C
2040 LET C = A*A + B*B                   Assigns 725 to variable C
```

In the last example, multiplication and addition were executed in the same statement. The multiplications were evaluated first and the additions were evaluated last. Multiplication and division are evaluated on a left to right basis with multiplication and division always occuring prior to addition and subtraction. This procedure can be altered by using parenthesis as shown below:

```
2060 LET C = A*A/(4*B)                   Assigns one to variable C
2070 LET C = A*A/4*B                     Assigns 625 to variable C
```

The last example multiplied the value of variable A by itself, then divided the result by four, and then multiplied the result by the value of variable B. The result in the previous example was different since the denominator was evaluated prior to the division.

2.2.8 NUMERICAL DECISIONS

The 'IF' statement allows decisions to be based upon numerical values of variables. The form is as follows:

135 IF A=10 THEN GOTO 1200 Branch to line #1200 if A = 10

or *145 IF A<>10 THEN* PRINT *"GOOD TEST"*

Any statement can be placed after the 'IF - THEN' sequence. Other operators can be used in conjunction with 'IF - THEN' as follows:

'<'	(less than)	'>'	(greater than)
'='	(equal)	'<>'	(not equal)
'<='	(less or equal)	'>='	(greater or equal)

A logical 'AND' can be created by the if statement as follows:

1000 IF A<10 THEN IF A>0 THEN GOTO 3000

This statement will branch to line #3000 if the number A has a value one through nine. Otherwise, it will appear not to execute and continue with the next statements in the program.

Another example of an 'AND' is as follows:

1000 IF A = 10 THEN IF B = 4 THEN GOTO 200

The branch will occur if A=10 and B=4.

A logical 'OR' can be created as follows:

1000 IF A <> 10 THEN IF B <> 4 THEN GOTO 200

This branch will not occur if A=10 or if B=4. The commands following the 'IF' statement will be executed if A=10 or B=4.

The word 'THEN' need not be included in the 'IF' statement, but it is shown for clarity.

2.2.9 IEEE BUS BUFFERS

The IEEE Bus is fully buffered on input and output. The input buffer stores up to forty characters received from an external device. The output buffer also stores up to forty characters to be sent out on the bus. The input buffer functions as shown below.

When data is received via a BUSIN command, it is stored in the input buffer. To display contents of the input buffer the following statement is used.

```
PRINT I$(1)
```

This displays the ASCII value in decimal of the first character of the input buffer. With this statement you can "read" the contents of the input buffer. The contents of the input buffer may also be assigned to variables within a program using the following:

```
A = I$(1)
```

In this example the content of the first character of the input buffer is assigned to variable A.

The contents of the input buffer may also be tested by using the numerical IF statement as follows.

```
IF I$(4)=65 THEN GOTO 2000
```

If the fourth character is an 'A' - goto line #2000

The use of a for-next loop can search for a specific character as follows:

```
1000 FOR I = 1TO40
1010 IF I $(I) = 65 J = I
1020 NEXT I
```

Test for an 'A' - Let J = I (J = position of last 'A')

To find first occurance of the letter 'A' the program is altered as follows:

```
1000 FOR I = 1TO40
```

```
1010 IF I$(I) = 65 J = I

1015 IF I$(I) = 65 I = 40          Let I = 40 when first 'A' is found
                                   in the data string.

1020 NEXT I                        Search stops since I = 40
```

This method of exit from a for-next loop is good programming practice since this procedure completes the for-next loop properly.

The output buffer is written into as follows:

```
1000 O$(1) = 65                    First character is an 'A'

1010 O$(2) = 66                    Second character is a 'B'

1020 BUS OUT                       Output buffer (two characters 'AB')
```

Line 1000 assigns an ASCII 65, or 'A' into the first character of the output buffer. The 1010 assigns an ASCII 66 or 'B' into the second character of the output buffer. When no text follows the BUSOUT statement, the output data will be taken from the output buffer. In the case shown above, the characters A and B will be output following the BUSOUT statement.

This buffer is initially filled with forty zeros which act as delimiters. When the value is changed to non-zero, the ASCII character equivalent to the decimal value will be sent on a BUSOUT execution. For this reason the buffer must be filled from the character #1 to the last character desired. The output string will be of the length equal to the number of characters entered. A zero in the buffer will terminate the string.

The output buffer is cleared by 'RUN', but is not cleared by BUSOUT statement. The buffer must be cleared or altered by the program.

This buffer allows non-ASCII characters to be sent over the bus and allows a string to be concatenated and then sent by a busout statement.

2.2.10 TABULATION

The tab function can be used to format PRINT statement output:

Examples:

```
1040 PRINT TAB(10); "TEST"          Tabs to position 10 - Prints
1050 PRINT TAB(10); "TEST";         Prints after position 10
1060 PRINT TAB(15); "THIS";         Prints second word
1070 PRINT TAB(20); "TEXT"          Prints last word
```

2.2.11 CART-INFO

The PROM cartridge can be examined by the CART-INFO verb. Depress CTRL followed by the 2 key. (This used the AUTO-VERB feature.) The display should be as follows:

```
CART-INFO
```

After you push RETURN , the display should show the following:

```
CART-INFO 404 USED,3 PROGRAMS,AUTO=1
```

This indicates that there are 404 bytes used in the cartridge, 3 programs stored in the cartridge, and that the AUTOSTART program is the first program in the cartridge.

2.2.12 SECONDARY ADDRESSING

The "BUSADDRESS" command has been augmented to allow the specification of an additional listen only address or a secondary address. The syntax is as follows:

```
BUSADDRESS 00001,  00010     Additional listen only
BUSADDRESS 00001;  0010      Secondary address
```

The primary address places the code (00100001) on the Bus if the IEEE-488 device is to be a listener, and the code (01000001) on the Bus when the device is to be

a talker. When an additional listen only address is specified two codes are placed on the Bus with ATN (attention) asserted. The first code is as above, followed by the code (001000010) to specify an additional listener, such as a printer. This code will not change when the primary address changes from listener to talker.

If a secondary address is specified two addresses are placed on the Bus as above, but the second address will be the code (01100010). This code will address a subset of the primary address if secondary is implemented in the device on the Bus.

If no second address is specified, only one address will be placed on the Bus. The address will appear only if a change in address occurred or if the Bus changes direction of transfer. An UNL (unlisten) is included with listen addresses.

2.2.13 BUSTIME SPECIFICATION

The "BUSIN" or "SERIAL POLL" mode of the Controller assumes that a device is on the Bus with the specified address and that the device will make the "DAV" (Data Valid) line true when it is to talk. If the unit is not present the system will wait for "DAV", causing the system to stop.

This can be avoided if desired by setting a software timer with the following syntax:

BUSTIME 1	Set timer for 100 ms (approximately)
BUSTIME 10	Sets timer for 1 second
BUSTIME 0	Sets timer for maximum (default)

The bustime timer will cause the exit from the "BUSIN" or "SERIAL POLL" mode and start the next BASIC statement. If the input buffer is tested as follows, the timeout condition can be detected:

```
IF I$(1) = 0 THEN PRINT "TIMEOUT"
```

The input buffer is cleared, and if no message is talked, the first character in the buffer will still be zero.

The bustime auto verb is CTRL (control) "5".

2.2.14 PROGRAM CHAINING

The "READ ROM" single stroke key which produces the display:

```
PROGRAM NUMBER?
```

Can be included in a basic program if followed by a number or a variable that has the numeric value of the desired program on the PROM cartridge.

Execution of this statement will cause the specified program to be loaded and started from the first program line. The variable values which have simple names (A through Z) will be maintained. Variables which are in arrays are not maintained. (The simple variables have fixed locations in memory. The arrays are stored in the text area and are destroyed.)

This feature allows the Model 3520 to have memory as large as the PROM cartridge. The introduction of the 4K PROM cartridge option makes this significant.

In "CHAINING" programs variables can be used as flags to indicate the status. The first program called the first time may have a special response required as opposed to the second or third time. This status can be determined as follows:

```
10 IF A=0 GOTO 1000
20 IF A=1 GOTO 2000
```

Since the simple variables are assigned the value of zero at power up, the first execution of the program will execute the code starting at line 1000. The value of the variable "A" can be changed to one by the "CHAINED" program such that the code starting at 2000 will be executed when the first program is executed the second time.

2.2.15 PROM ERASER ACCESSORY

The PROM Eraser Accessory can be added to any Model 3520 which has a rear cover plate to the left of the transformer. The Power connector for the PROM Eraser is attached to the back side of this cover plate.

To install the PROM Eraser, remove the plate (saving screws) and cut the cable loose from the plate, plug in cable to PROM Eraser unit and install into the opening in the panel and retain with the screws.

CAUTION

This unit contains an Ultraviolet source which can be harmful to the eyes. The best practice is not to start lamp until the PROM module is in place and allow timer to turn off lamp before removing the PROM module.

When the PROM is to be erased, remove cover and install into the eraser with the "window" on the PROM pointing down. Push switch for several seconds to warm up lamp, and release. The indicator lamp should remain on. After 30 minutes the lamp will be turned off automatically. The PROM is now erased and ready for reuse as a program storage medium.

The full erase time should be used to assure full erasure. The power to the BUSser must be maintained during this interval.

Interuption of power will turn lamp off, restarting will retime the full interval. Excessive erase time causes no bad effects on the PROM.

If the lamp is to be turned off prior to the 30 minute interval, press switch for a short interval, or turn off main power.

2.3 ERROR DICTIONARY

2.3.1 Program Error Messages

Error Number	Probable Cause
1	Unrecognizable Statement
2	Numeric Overflow
3	Illegal Character of Variable
4	Mismatch of Quote Marks
5	Dimension Error
6	Illegal Arithmetic
7	No Such Line Number
8	Divide by Zero Executed
9	Subroutines Nested More Then Eight Deep
10	Return Without Gosub
11	Illegal Variable
12	Syntax Error
13	Parenthesis Error
14	Memory Overflow
15	Subscript Error
16	For-Next Loops Nested More Than Eight Deep
17	Next Without For

2.3.2 IEEE Buffer Errors

Error Number	Probable Cause
18	Input Buffer - Index Out of Range
19	Output Buffer - Value Out of Range
	or
	Index Out of Range (1 to 40)
20	Busaddress Value Error

2.3.3 Bus Error Messages

Error Number	Probable Cause
21	BUSIN - No Numeric Data for Variable
22	BUSIN - More Variable Names than Data
23	BUSIN - More Data Than Variable Names
24	ATN - Value Out of Range (0 to 255)

2.3.4 ROM Error Messages

Error Number	Probable Cause
30	Program ROM - No Blank ROM Space
31	Program ROM - Unerased Byte(s) in area to be Used for Saving Program
	or
	Program ROM - No Text to Save
32	Program ROM - Insuficient Space
33	Program ROM - ROM Verify Error (Text was Stored, Failed Verify)
35	Load ROM - ROM Number Zero
36	Load ROM - Specified Program Not Found
37	Load ROM - Check Sum Error (Program was Loaded into Text Area)
38	Auto-Start Program Number Error
39	Undefined Function Not Present

CHAPTER 3

REPLACEABLE PARTS

3-1. INTRODUCTION

This section contains the spare parts lists for each assembly in the Model 3520 Instrumentation Controller. Chapter 3 shows the assemblies where the listed parts are installed. Ordering information is included and should be used when purchasing parts from the factory. Explanation of the column listings and Table 3-1 Manufacturers Code-to-Name Index are also included.

3-2. PART ORDERING INFORMATION

To order a part from the Replaceable Parts List, state the Systron-Donner part number, indicate the quantity required and address the order to the nearest Systron-Donner Sales or Service Center.

To order a part not listed in the Replaceable Parts List, state the instrument model number, instrument serial number, the description and function of the part, and the number of parts required. Address the order to the nearest Systron-Donner Sales or Service Center.

3-3. PARTS LISTS

Table 3-1 gives the Code-to-Name information for the manufacturers listed in the replaceable parts lists. The replaceable parts lists are placed in A numbered sequence.

Each column in the parts lists provide specific information relating to the listed parts as follows:

1) DESIGNATOR: Reference numbers applicable to assemblies are listed in alpha-numeric order.

2) COMPONENT NOMENCLATURE AND DESCRIPTION: Component names and specifications are provided in these columns and include: value, tolerance, wattage rating, working voltage, construction, etc.

3) MANUFACTURER'S CODE: H4-2 Federal Supply Code numbers are listed to identify component manufacturers. A Manufacturers Index is provided in this section for cross-reference.

4) MANUFACTURER's PART NUMBER: True manufacturer part numbers are listed in this column.

5) SYSTRON-DONNER STOCK NUMBER: The Systron-Donner stock number is listed in this column.

6) T/Q (Total Quantity): The quantity, appearing after an item entry, indicates the number of times the component is used in that assembly. The total quantity for each part is given only once -- at the first appearance of the part number in the list.

TABLE 3-1 MANUFACTURER'S CODE-TO-NAME INDEX

Code	Name & Address
00809	Croven Ltd. 500 Beech Street Whitby Ontario, Canado
01121	Allen-Bradley Co. 1201 So. 2nd Street Milwaukee, WI 53204
01295	Texas Instruments Inc. Semiconductor & Components Div. 13500 No. Central Expressway Dallas, TX 75231
02660	Bunker Ramo Corp. Amphenol Connector Division 2801 So. 25th Avenue Broadview, IL 60153
02735	RCA Corp. Solid State Division Route 202 Somerville, NJ 08876
03296	Nylon Molding Corp. 40 Brown Street Springfield, NJ 07081
03508	General Electric Co. Semiconductor Products Dept. Electronics Park Syracuse, NY 13201
03877	Transitron Electronic Corp. 168 - 186 Albion Street Wakefield, MA 01880
03888	Pyrofilm Corp. 60 So. Jefferson Road Whippany, NJ 07981
04713	Motorola Semiconductor Prod. Inc. 5005 E. McDowell Road Phoeniz, AZ 85008
05245	Corcom Inc. 2857 No. Halsted Street Chicago, IL 60657
05574	Viking Industries Inc. 21001 Nordhoff Chatsworth, CA 91311
06383	Panduit Corp. 17301 Ridgeland Tinley Park, IL 60477
06540	Amatom Electronic Hardware Division of Mite Corp. 81 Rockdale Avenue New Rochelle, NY 10806
06776	Robinson Nugent Inc. P.O. Box 470 New Albany, IN 47150
07047	The Ross Milton Co. 511 Second Street Pike Southampton, PA 18966
07263	Fairchild Camera & Instrument Corp. Semiconductor Division 313 Frontage Road Mountain View, CA 94040
08353	C and K Components Inc. 103 Morse Newton, MA 02158
11236	CTS of Berne Inc. 406 Parr Road Berne, IN 46711
12014	Chicago Rivet & Machine Co. 950 So. 25th Avenue Bellwood, IL 60104
12697	Clarostat Mfg. Co. Inc. Lower Washington Street Dover, NH 03820
15636	Elec-Trol Inc. 26477 No. Golden Valley Road Saugus, CA 91350
16299	Corning Glass Works Electronic Components Division 3900 Electronics Drive Raleigh, NC 27604
18324	Signetics Corp. 811 E. Arques Sunnyvale, CA 94086
19701	Mepco/Electra Inc. Electra Division P.O. Box 760 Mineral Wells, TX 76067
21604	The Buckeye Stamping Co. 555 Marion Road Columbus, Ohio 43207

TABLE 3-1 MANUFACTURER'S CODE-TO-NAME INDEX (Cont'd)

Code	Name & Address
24335	Cooper Laboratories Inc. Mystic, CT
24796	Parelco Inc. P.O. Box 116 San Juan Capistrano, CA 92675
27014	National Semiconductor Corp. 2950 San Ysidro Way Santa Clara, CA 95051
27264	Molex Products Co. 5224 Katrine Avenue Downers Grove, IL 60515
27556	IMB Electronic Prod. Inc. 15401-E Carmenita Road Santa Fe Springs, CA 90670
28520	Heyman Mfg. Co. 147 N. Michigan Avenue Kenilworth, NJ 07033
30146	A P Inc. 72 Corwin Drive Painesville, OH 44077
31918	International Electro Exchange Inc. 8081 Wallace Road Eden Prairie, NM 55343
50522	Monsanto Co. Electronic Special Products 10131 Bubb Rd. Cupertino, CA 95014
51495	R & D Plastics 780 - 32nd Street Richmond, CA 94804
51642	Centre Engineering Inc. P.O. Box P-8 State College, PA 16801
52542	Systron-Donner Corp. Instrument Division 10 Systron Drive Concord, CA 94518
56289	Sprague Electronic Co. North Adams, MA 01247
71279	Cambridge Thermonic Corp. 445 Concord Avenue Cambridge, MA 01238
71400	Bussmann Mfg. Division of McGraw-Edison Co. 2536 W. University Street St. Louis, MO 63017
71590	Globe-Union Inc. Centralab Electronics Division 5757 No. Green Bay Avenue Milwaukee, WI 53201
71785	TRW Electronic Components Cinch Connector Operations 1501 Morse Avenue Elk Grove Village, IL 60007
73138	Beckman Instruments Inc. Helipot Division 2500 Harbor Blvd. Fullerton, CA 92634
73734	Federal Screw Products Inc. 3917 N. Kenzie Avenue Chicago, IL 60618
73957	Groov-Pin Corp. 1125 Hendricks Causeway Ridgefield, NJ 07657
75915	Littlefuse Inc. 800 E. Northwest Hwy. Des Plaines, IL 60016
77342	AMF Inc. Potter & Bumfield Division P.O. Box 522 - 1200 E. Broadway Princeton, IN 47570
78189	Illinois Tool Works Inc. Shakeproof Division St. Charles Road Elgin, IL 60120
79963	Zierick Mfg. Co. Radio Circle Mt. Kisco, NY 10549
80294	Bourns Inc. Instrument Division 6135 Magnolia Avenue Riverside, CA 91506
81073	Grayhill Inc. La Grange, IL 60525

TABLE 3-1 MANUFACTURER'S CODE-TO-NAME INDEX (Cont'd)

Code	Name & Address
81349	Mil Std
81483	International Rectifier Corp. 9220 Sunset Blvd. Los Angeles, CA 90069
81831	Filtron Co. Inc. Flushing Long Island, NY
82389	Switchcraft Inc. 5555 N. Elston Avenue Chicago, IL 60630
83330	Herman H. Smith Inc. 812 Snediker Avenue Brooklyn, NY 11207
86928	Seastrom Mfg. Co. Inc. 701 Sonora Avenue Glendale, CA 91201
88245	Litton Systems Inc. USECO Division 13536 Saticoy Street Van Nuys, CA 91409
90201	Mallory Capacitor Co. P.O. Box 372 – Washington Street Indianapolis, IN 46206
91418	Radio Materials Co. 4242 W. Bryn Mawr Chicago, IL 60646
91506	Augat Inc. 33 Perry Avenue Attleboro, MA 02703
91637	Dale Electronics Inc. P.O. Box 609 Columbus, NE 68601
95146	ALCO Electronic Prod. Inc. P.O. Box 1348 Lawrence, MA 01842
96733	San Fernando Electric Mfg. Co. 1509 First Street San Fernando, CA 91341
96906	Mil Std
98978	International Electronic Research Corp. 135 W. Magnolia Avenue Burbank, CA 91502
99392	S T M Corp. 2904 Chapman Street Oakland, CA 94601

REPLACEABLE PARTS LIST

TITLE		FINAL ASSEMBLY #07362001 Rev A_4				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
6		CABINET, Front	52542	073641	073641	1
7		CABINET, Rear	52542	073625	073625	1
8		NUT, Speed 6-32	78553	C1029355-632-24	101123	2
9		SCREW, PHMS 6-32 x 3/8	96906	MS51957-28	10063206	2
10		NUT, Kep 6-32	78189	511-061800-01	112472	6
11		ASSEMBLY, Chassis	52542	07362101	07362101	1
13		WINDOW	52542	07362901	07362901	1
14		SCREW, Black Oxide 6-32 x 3/8	08547	By Description	102775	5
15		WASHER, Black Oxide	52542	62407190	62407190	2
16		PLATE, Cover	52542	073721	073721	5
17		WASHER, Flat #6	96906	MS15795-805	100709	2
18		WASHER, Split Lock #6	96906	MS35338-136	100712	2
19		LABEL	52542	03711201	03711201	1

REPLACEABLE PARTS LIST

TITLE CHASSIS ASSEMBLY #07362101 Rev B_2

ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
8		GUSSET, Right	52542	07363801	07363801	1
9		GUSSET, Left	52542	07363802	07363802	1
10		NUT, Speed 6-32	78553	C10293SS-632-24	101123	14
12	A2	ASSEMBLY, Keyboard PC	52542	07364001	07364001	1
13		GROMMET	52238	PGS-2	117739	2
15	A1	ASSEMBLY, Rear Panel	52542	07363401	07363401	1
16		SPACER, 6-32	02660	552633-4	116491	1
17		NUT, Hex 4-40	96906	NAS671C4	100622	2
18		BRACKET, Display & Window	52542	073639	073639	1
19	A3	ASSEMBLY, Display Board PC	52542	07362301	07362301	1
20		SPACER, 6-32	06540	8220B-0632-3A	100502	2
21		SCREW, FHMS Black 6-32			116336	2
22	A4	ASSEMBLY, Processor PC	52542	07362401	07362401	1
23		SPACER, Jackscrew	52542	073673	073673	2
25		PLATE, Bottom	52542	073635	073635	1
26		FOOT, Fubber Black	77969	4460	103880	4
28		LUG, Solder #8	78189	2104-08-00	102961	1
31		PUSHBUTTON, Key BUSOUT	52542	07364701	07364701	1
32		PUSHBUTTON, Key BUSIN	52542	07364702	07364702	1
33		PUSHBUTTON, Key BUSADD	52542	07364703	07364703	1
34		PUSHBUTTON, Key BUSCLR	52542	07364704	07364704	1
35		PUSHBUTTON, Key BUSREM	52542	07364705	07364705	1
36		PUSHBUTTON, Key BUSLOC	52542	07364706	07364706	1
37		PUSHBUTTON, Key BUSPOLL	52542	07364707	07364707	1
38		PUSHBUTTON, Key BUSLIST	52542	07364708	07364708	1
39		PUSHBUTTON, Key PROGROM	52542	07364709	07364709	1
40		PUSHBUTTON, Key READROM	52542	07364710	07364710	1
41		PUSHBUTTON, Key BUSPRINT	52542	07364711	07364711	1
42		PUSHBUTTON, Key BUSSQR	52542	07364712	07364712	1
45		PUSHBUTTON, Key PRINT	52542	07364801	07364801	1
46		PUSHBUTTON, Key !/1	52542	07364802	07364802	1
47		PUSHBUTTON, Key "/2	52542	07364803	07364803	1
48		PUSHBUTTON, Key #/3	52542	07364804	07364804	1
49		PUSHBUTTON, Key $/4	52542	07364805	07364805	1
50		PUSHBUTTON, Key %/5	52542	07364806	07364806	1
51		PUSHBUTTON, Key &/6	52542	07364807	07364807	1
52		PUSHBUTTON, Key '/7	52542	07364808	07364808	1
53		PUSHBUTTON, Key (/8	52542	07364809	07364809	1
54		PUSHBUTTON, Key)/9	52542	07364810	07364810	1
55		PUSHBUTTON, Key ¢	52542	07364811	07364811	1
56		PUSHBUTTON, Key */:	52542	07364812	07364812	1
57		PUSHBUTTON, Key =/-	52542	07364813	07364813	1
58		PUSHBUTTON, Key RUN	52542	07364814	07364814	1

REPLACEABLE PARTS LIST

TITLE		CHASSIS ASSEMBLY #07362101 (Cont'd)				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
59		PUSHBUTTON, Key LIST	52542	07364815	07364815	1
60		PUSHBUTTON, Key Q	52542	07364816	07364816	1
61		PUSHBUTTON, Key W	52542	07364817	07364817	1
62		PUSHBUTTON, Key E	52542	07364818	07364818	1
63		PUSHBUTTON, Key R	52542	07364819	07364819	1
64		PUSHBUTTON, Key T	52542	07364820	07364820	1
65		PUSHBUTTON, Key Y	52542	07364821	07364821	1
66		PUSHBUTTON, Key U	52542	07364822	07364822	1
67		PUSHBUTTON, Key I	52542	07364823	07364823	1
68		PUSHBUTTON, Key ←/O	52542	07364824	07364824	1
69		PUSHBUTTON, Key @/P	52542	07364825	07364825	1
70		PUSHBUTTON, Key STEP	52542	07364826	07364826	1
71		PUSHBUTTON, Key RETURN	52542	07364827	07364827	1
72		PUSHBUTTON, Key CTRL	52542	07364828	07364828	1
73		PUSHBUTTON, Key A	52542	07364829	07364829	1
74		PUSHBUTTON, Key S	52542	07364830	07364830	1
75		PUSHBUTTON, Key D	52542	07364831	07364831	1
76		PUSHBUTTON, Key F	52542	07364832	07364832	1
77		PUSHBUTTON, Key G	52542	07364833	07364833	1
78		PUSHBUTTON, Key H	52542	07364834	07364834	1
79		PUSHBUTTON, Key J	52542	07364835	07364835	1
80		PUSHBUTTON, Key [/K	52542	07364836	07364836	1
81		PUSHBUTTON, Key /L	52542	07364837	07364837	1
82		PUSHBUTTON, Key +/;	52542	07364838	07364838	1
83		PUSHBUTTON, Key CHAR DEL	52542	07364839	07364839	1
84		PUSHBUTTON, Key STOP	52542	07364840	07364840	1
85		PUSHBUTTON, Key SHIFT	52542	07364841	07364841	1
86		PUSHBUTTON, Key Z	52542	07364842	07364842	1
87		PUSHBUTTON, Key X	52542	07364843	07364843	1
88		PUSHBUTTON, Key C	52542	07364844	07364844	1
89		PUSHBUTTON, Key V	52542	07364845	07364845	1
90		PUSHBUTTON, Key B	52542	07364846	07364846	1
91		PUSHBUTTON, Key ↑/N	52542	07364847	07364847	1
92		PUSHBUTTON, Key]/M	52542	07364848	07364848	1
93		PUSHBUTTON, Key </,	52542	07364849	07364849	1
94		PUSHBUTTON, Key >/.	52542	07364850	07364850	1
95		PUSHBUTTON, Key ?//	52542	07364851	07364851	1
96		PUSHBUTTON, Key Space Bar	52542	07364852	07364852	1

REPLACEABLE PARTS LIST

TITLE		A1, POWER SUPPLY ASSEMBLY #07363401 Rev E				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
5		PANEL, Rear	52542	073634	073634	1
7		CORD, Line	52542	T921000	T921000	1
8		STRAIN Relief	52542	T930000	T930000	1
10	XF1	POST, Fuse	75915	345001	117377	1
11	F1	FUSE, S-B 3AG 1/2A	71400	MDL 1/2	100589	1
13	S1	SWITCH, Slide 115/230V	82389	46256LFR	102331	1
15	S2	SWITCH, Toggle DPDT	09353	7201PN	102354	1
17	T1	TRANSFORMER, Power	52542	07366601	07366601	4
18		SCREW, PHMS 6-32 x 1-1/2	96906	MS51957-36	10063224	1
19	TS1	STRIP, Terminal	71785	2008	103035	1
20		WASHER, Fiber #6	86445	S7-Black	100619	4
21	CR1	DIODE, Rectangular 400V	04713	1N5395	100413	2
	CR2	Same as CR1				
22	CR3	DIODE, Rectifier 50V	81483	30S05	103252	2
	CR4	Same as CR3				
24	XU1	SOCKET, TO-3			830070	1
25	U1	INTEGRATED CIRCUIT Voltage Regulator	27014	LM309K	025787	1
26	C1	CAPACITOR, Ceramic .1 µF 20%, 50V	56289	5C023104X0500C5	100178	1
29	C2	CAPACITOR, El Can 4600 µF +70-10%, 15V	56289	36D451G100AA2A	116493	1
30	C3	CAPACITOR, El Can 450 µF, 100V	56289	36D451G100AA2A	116493	1
32		CLAMP, Cap 1-3/8	56289	CM5-22	100725	2
35	J15,J16	CONNECTOR, Shell 5-pos	27264	03-06-1055	101264	2
37	J18	CONNECTOR, Shell 5-pos	27264	03-06-2055	101263	1
38	E2	LUG, Solder #6	73734	1955	100490	1
39	E1	LUG, Solder #8	78189	2104-08-00	102961	1
42		JACK, Red	74970	108-1102-001	101275	1
44		RESISTOR, CC 22 Ω 5%, 2W	01121	HB2205	101655	1

REPLACEABLE PARTS LIST

TITLE	A1, POWER SUPPLY ASSEMBLY #07363401 (Cont'd)					
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
46		HARNESS, Cable	52542	07370001	07370001	1
47		WIRE, Terminal	52542	T960565	T960565	4
48		CONNECTOR, Terminal Female	27264	02-06-11-3	10282201	9
49		TIE, Cable	06383	SSTIM	100753	38
51		SCREW, PHMS 4-40 x 5/16	96906	MS51957-14	10062605	2
52		SCREW, PHMS 4-40 x 3/8	96906	MS51957-15	10062606	2
54		WASHER, Flat #4	96906	MS15795-803	100703	2
55		WASHER, Kep #4	78189	511-041800-00	100941	4
56		SCREW, PHMS 6-32 x 7/16	96906	MS51957-29	10063207	2
57		SCREW, PHMS 6-32 x 3/8	96906	MS51957-28	10063206	10
58		SCREW, PHMS 6-32 x 9/16	08547	By Description	100851	2
59		WASHER, Flat #6	86928	5710-23-10	100662	8
60		NUT, Kep 6-32	78189	511-061800-01	112472	8
61		NUT, Kep 6-32 Large	78189	511-061800-00	100862	4
62		WASHER, Split-Lock #6	96906	MS35338-136	100712	6
63		SCREW, PHMS 8-32 x 5/16	96906	MS51957-42	10064205	1
64		NUT, Kep 8-32	78189	511-081800-00	100940	1
66		PLATE, Cover Eraser PROM	52542	073681	073681	1

REPLACEABLE PARTS LIST

TITLE		A2, KEYBOARD PC ASSEMBLY #07364001 Rev B				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	C1	CAPACITOR, Tant 39 µF 10%, 10 V	56289	150D396X9010B2	100183	2
	C2	Same as C1				
	C3 thru C7	CAPACITOR, Ceramic .1 µF +80 -20%, 10 V	71590	6K10-104	100120	5
	J2	CONNECTOR, Rt. Angle	02660	552791-1	116490	1
	J10	CABLE, Interconnect	52542	07363601	07363601	1
	P14	CONNECTOR, PC Mtg. 36-pin	71785	57-10360-13	116509	1
	R1 thru R4	RESISTOR, CC 100 k 5%, 1/4 W	01121	CB1045	10155801	4
	R5	RESISTOR, CC 100 k 5%, 1/2 W	01121	EB2225	101454	1
	U1 thru U5	INTEGRATED CIRCUIT, CMOS BCD-DEC DCD	04713	MC14028BCP	045286	5
	U6	INTEGRATED CIRCUIT, CMOS 4520 Dual 4 Bit Upcntr	04713	MC14520BCP	117278	2
	U7	Same as U6				
	U8	INTEGRATED CIRCUIT, CMOS 4518 Dual BCD Upcntr	04713	MC14518BCP	103339	1
	U9	INTEGRATED CIRCUIT, MOS 145562 Bin 1/4	04713	MC6821P	116281	3
	U10 thru U12	INTEGRATED CIRCUIT, MOS Program Periph Inter	04713	MC14556BCP	116492	1
	U13 thru U15	Not Used				
	U16 thru U17	INTEGRATED CIRCUIT, TTL MSI 488 8 x CDR	04713	MC3447P3	117956	2
	U18	INTEGRATED CIRCUIT, MOS Gen'l Purpose Intrfc A	04713	MC68488P	116430	1
		CABLE, Flat Jumper	15912	FSN-23A-20	116488	4
		KEYBOARD, 60-Pos		S-60-C-0000	116504	1
		SPACER, Swage 6-32	88245	1530 B 3/8	100533	4

REPLACEABLE PARTS LIST

TITLE	A2, KEYBOARD PC ASSEMBLY #07364001 (Cont'd)				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		SCREW, 4-40 x 5/16	73734	By Description	100821	18
		WASHER, Fiber #4	83330	2161	100670	16
		SWITCH, Module	52542	07364601	07364601	3
		BOARD, PC	52542	073640	073640	1

REPLACEABLE PARTS LIST

TITLE	A2, INTERCONNECT CABLE ASSEMBLY #07363601 Rev A				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
7		CONNECTOR, PCB 34-pin	75037	3402-000T	116494	1
10		CABLE, Flat 34-pin	76381	3365/34	116495	1
13		CONNECTOR, 34-pos	53387	3414-3000	116134	1

REPLACEABLE PARTS LIST

TITLE	A3, DISPLAY PC ASSEMBLY #07362301 Rev C					
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	C1	CAPACITOR, Ceramic .1 µF +80-20%, 10V	71590	UK10-104	100120	1
	C2	CAPACITOR, E1 Ax 10 µF, 100V	56289	TE1407	106449	1
	CR1	LED, Green	28480	HLMP-2550	116486	2
	CR2	Same as CR1				
	P16	CONNECTOR, Rectangular 5-pos	27264	03-06-2055	101263	1
	Q1	TRANSISTOR, ISNPN	04713	2N5190	101394	2
	Q2	TRANSISTOR, SIPNP	04713	2N5193	101405	1
	Q3	Same as Q1				
	R1	RESISTOR, CC 10 k 5%, 1/4W	01121	CB1035	10157001	1
	R2	RESISTOR, CC 1 k 5%, 1/4 W	01121	EB1025	101467	1
	R3	RESISTOR				
	R4	RESISTOR, CC 1.2 k 5%, 1W	01121	GB1225	101506	2
	R5	Same as R4				
	U1 thru U10	INTEGRATED CIRCUIT, LIN MSC 60V 8NPN Darlington	56289	UDN-6118A-2	116483	10
	U11	INTEGRATED CIRCUIT Voltage Regulator	04713	MC78M20CT	116506	1
	V1	DISPLAY, 40-Digit		DC405A2	116487	1
	VR1	DIODE, Zener 18V 5%	04713	1N967B	100404	1
		HEATSINK, (for Q2)	98978	PA1-1CB	100978	1
		SOCKET, IC 16-pin	73803	C91-16-50	102927	1
		SOCKET, IC Terminal	27264	05-33-0020	116484	4
		CONNECTOR, Terminal Male	27264	02-06-2103	10282202	3
		TIE, Cable	06383	SSTIM	100753	A/R
		SOCKET, IC	27264	06-05-3201	116485	4
		SCREW, PHMS 4-40 x 7/16	96906	MS51957-16	10062607	1
		WASHER, CPRSN	04713	B52200F003	103351	1
		NUT, Hex 4-40	86928	5710-23-10	100662	1
		BOARD, PC	52542	073623	073623	1

REPLACEABLE PARTS LIST

TITLE	A4, PROCESSOR PC ASSEMBLY #07362401 Rev B_4					
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	C1	CAPACITOR, Tant 39 µF 10%, 10 V	56289	150D396X9010B2	100183	2
	C2	Same as C1				
	C3 thru C11	CAPACITOR, Ceramic .1 µF +80 -20%, 10 V	71590	UK10-104	100120	9
	C13	CAPACITOR, DM 18 pF 5%, 500 V	72136	DM10CD180JO	100128	2
	C14	Same as C13				
	C15	CAPACITOR, Tant 22 µF 10%, 15 V	56289	150D226X9015B2	100125	1
	CR1	DIODE, Signal 50 V	03508	1N4151	100385	1
	P10	CONNECTOR, 34-pin	53387	3431-1002	116133	1
	P15	CONNECTOR, Rectangular 5-pos	27264	03-06-2055	101263	1
	R1 thru R3	RESISTOR, CC 3 k 5%, 1/4 W	01121	CB3025	10154801	3
	R4	RESISTOR, CC 22 k 5%, 1/4 W	01121	CB2235	10157201	1
	S1	SWITCH, PB	31918	MDP W/AG CONT	117899	1
	U1	INTEGRATED CIRCUIT, MOS MPU W/Clock and RAM	04713	MC6802P	116285	1
	U2	INTEGRATED CIRCUIT, TTL SSI 4 x 2 NAND	01295	SN74132N	045270	1
	U3 thru U6	INTEGRATED CIRCUIT, TTL MSI Dual 2T04	01295	SN74LS155N	116116	4
	U7 thru U9	INTEGRATED CIRCUIT, MOS PROM 2 k * 8 uv ers	34649	B2716	116440	3
	U10	Not Used				
	U11 thru U14	INTEGRATED CIRCUIT	34649	P2114L	116303	4
	XU1	SOCKET, IC 40-pin	73803	C83-40-02	117159	1
	XU7-9	SOCKET, IC 24-pin	73803	C84-24-02	103845	3
	Y1	OSCILLATOR, Crystal 4 MHz	75378	MP040	116286	1

REPLACEABLE PARTS LIST

TITLE	A4, PROCESSOR PC ASSEMBLY #07362401 (Cont'd)				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		KEY. Polarizing (P10 P10)	53387	3518	11613301	2
		TIE, Cable	06383	SSTIM	100753	A/R
		CONNECTOR, Terminal Male	27264	02-06-2103	10282202	2
		BOARD, PC	52542	073624	073624	1

REPLACEABLE PARTS LIST

TITLE OPTION 001; PROM 2k READER/PROGRAMMER #07366801 REV B

ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		DELETE				
		PLATE, Cover	52542	073721	073721	1
		ADD				
		ASSEMBLY, Cable PROM	52542	07364301	07364301	1
		SPACER, Brass Cad 6-1/8 x 1/4	83330	8701	100015	2
		MODULE, PROM (Ship with)	52542	07363001	07363001	1
		WIRE, 18 Awg PVC Grn 6"				

REPLACEABLE PARTS LIST

TITLE		OPTION 001; PROM 2k READER/PROGRAMMER #07364301 Rev A				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		CONNECTOR, Rect. 36-pin Male	15912	609-36M	118018	1
		CONNECTOR, Rect. 36-pin Fem.	15912	609-36F	118019	1
		CABLE, Flat 9"	15912	171-36	117977	1

REPLACEABLE PARTS LIST

TITLE		OPTION 001, PROM MODULE 2 k ASSEMBLY #07363001 Rev B				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		ASSEMBLY, PROM Module PC	52542	07369801	07369801	1
8		SHELL	52542	07367101	07367101	1
9		SCREW, PHMS 2-56 x 3/8	96906	MS51957-5	10062506	2
11		WASHER, S-L #2	96906	MS35337-77	100661	2
12		NUT, Hex 2-56	96906	MS35649-224	100636	2
13		MODULE, Cap	52592	07367102	07367102	1
15		LABEL, PROM ID	52542	073630	073630	1

REPLACEABLE PARTS LIST

TITLE		OPTION 001, 2k PROM MODULE PC ASSEMBLY #07363701 Rev A				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	C1	CAPACITOR, Ceramic .1 µF +80-20%, 10V	71590	UK10-104	100120	1
	CR1	LAMP, LED Red	28480	5082-4887	100460	1
	J14	CONNECTOR, Receptacle 36 pos			116580	1
	R1	RESISTOR, CC 10 k 5%, 1/4W	01121	CB1035	10157001	2
	R2	Same as R1				
	R3	RESISTOR, CC 417 k 5%, 1/4W	01121	CB4725	10159801	1
	U1	INTEGRATED CIRCUIT, MOS PRO MOS PROM	36349	B2716	116440	1
	W1,2	JUMPER	55210	L-2007-1	102879	2
		BOARD, PC	52542	073637	073637	

REPLACEABLE PARTS LIST

TITLE: OPTION 001, PROM MODULE 4k ASSEMBLY #0736002 Rev A

ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		ASSEMBLY, PROM Module PC	52542	07369802	07369802	1
8		SHELL	52542	07367101	07367101	1
9		SCREW, PHMS 2-56 x 3/8	96906	MS51957-5	10062506	2
10		WASHER, Flat #2	96906		116581	2
11		WASHER, S-L #2	96906	MS35337-77	100661	2
12		NUT, Hex 2-56	96906	MS35649-224	100636	2
13		MODULE, Cap	52592	07367102	07367102	1
15		LABEL, PROM ID	52542	073748	073748	1

REPLACEABLE PARTS LIST

TITLE	OPTION 001, PROM MODULE 4 k PC ASSEMBLY #07369802					
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	C1	CAPACITOR, Ceramic .1 µF +80-20%, 10V	71590	UK10-104	100120	1
	CR1	LAMP, LED Red	28480	5082-4887	100460	1
	CR2	DIODE, Signal 20V	03508	1N4151	100385	2
	CR3	Same as CR2				
	J14	CONNECTOR, Receptacle 3-pos			116580	1
	Q1	TRANSISTOR, SINPN	04713	2N3904	101377	2
	Q2	TRANSISTOR, SIPNP	04713	2N3906	101378	1
	Q3	Same as Q1				
	R1	RESISTOR, CC 10 k 5%, 1/4W	01121	CB1035	10157001	2
	R2	RESISTOR, CC 10 k 5%, 1/8W	01121	BB1035	101697	3
	R3	RESISTOR, CC 4.7 k 5%, 1/4W	01121	CB7425	10159801	1
	R4	Same as R2				
	R5	Same as R2				
	R6	Same as R1				
	R7	RESISTOR, CC 1 k 5%, 1/8W	01121	BB1025	101711	1
	W1	JUMPER, Molded	55210	L2007-1	102879	2
	W2	Same as W1				
		BOARD, PC	52542	073698	073698	1

REPLACEABLE PARTS LIST

TITLE		CONTROL LINE STATUS INDICATOR ASSEMBLY #07366301 Rev B_1				
ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		ASSEMBLY, Circuit Board PC	52542	07369901	07369901	1
9		SHELL, Modified	52542	07370901	07370901	1
12		CAP, Modified	52542	07371001	07371001	1
15		PANEL, Status	52542	07367104	07367104	1
17		PLUG, Red	74970	108-1002-001	101278	1
18		WASHER, Split-Lock #2	96906	MS35337-77	100661	2
19		SCREW, PHMS 2-56 x 3/8	96906	MS51957-5	10062506	2
20		NUT, Hex 2-56	96906	MS35649-224	100636	2
21		SCREW, 4-40 x 5/16	73734	By Description	100821	4
23		LABEL	52542	073749	073749	1

REPLACEABLE PARTS LIST

TITLE CIRCUIT BOARD PC ASSEMBLY #07369901 Rev A

ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	CR1 thru CR8	LAMP, LED Red	28480	HPA3082-4480	100460	8
	R1 thru R8	RESISTOR, CC 510 Ω 5%, 1/4 W	01121	CB5115	10161601	8
		PLUG, Strip Line	52542	073674	073674	1
	P2	CONNECTOR, 24-pin	02660	57-10240-3	116552	1

REPLACEABLE PARTS LIST

TITLE PROM ERASER ASSEMBLY #073701 Rev A

ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
		ASSEMBLY, PROM Eraser PC	52542	0736501	0736501	1
9		BASE, PROM Eraser	52542	073677	073677	1
11		COVER, PROM Eraser	52542	073678	073678	1
13		LIGHT SHIELD, PROM Eraser	52542	073679	073679	1
15		SCREW, FHMS 4-40 x 25	52542	MS24693-C2	1072004	4
16		SCREW, PHMS 4-40 x 25	52542	MS51957-13	10062604	3
17		WASHER, Split Lock #4	52542	MS35338-15	100711	3
18		WASHER, MICA #4	52542	MW500-125-002	112003	2
19		BUSHING, LED Clear	50522	MP51	1032002	1
24		MODULE, Cap Modified	52592	07371002	07371002	1
26		SCREW, 4-40 x 5/16	73734	By Description	100821	2
27		WASHER, Mylar #4	86928	5622-9-7	100921	2
29		THERMAL, Compound				A/R
31		LABEL, PROM Eraser	52542	073751	073751	1

REPLACEABLE PARTS LIST

TITLE PROM ERASER PC ASSEMBLY #07367501 Rev B

ITEM	REF	COMPONENT DESCRIPTION	MFR'S CODE	MANUFACTURER'S PART NUMBER	SD STOCK NUMBER	T/Q
	C1	CAPACITOR, Ceramic .01 µF +80-20%, 100V	91418	TA110	100103	1
	C2	CAPACITOR, E1 Ax 100 µF, 25V	56289	B41283-B5107-T	100165	1
	CR1	DIODE, Rectangular 400V	04713	1N5395	100413	1
	CR2	DIODE, Signal 50V	03508	1N4151	100385	2
	CR3	Same as CR2				
	CR4	LAMP, LED Red	50522	MV5753	117227	1
	DS1	LAMP, Ultra-Violet		BLE-2537S	116558	1
	P18	CONNECTOR, Rectangular 5-pos	27264	03-06-1055	101264	1
	Q1	TRANSISTOR, SIPNP 60V	04713	2N2905A	705067	1
	Q2	TRANSISTOR, SINPN 60V	04713	MJE800	106617	2
	Q3	Same as Q2				
	R1	RESISTOR, CC 10 k 5%, 1/4W	01121	CB1035	10157001	2
	R2	RESISTOR, CC 1 k 5%, 1/4W	01121	CB1025	10156901	1
	R3	RESISTOR, CC 20 Ω 5%, 1/2W	01121	EB2005	106009	2
	R4	Same as R3				
	R5	RESISTOR, CC 100 Ω 5%, 1/4W	01121	CB1015	10160901	1
	R6	Same as R1				
	S1	SWITCH			116568	1
	U1	INTEGRATED CIRCUIT, CMOS 12-bit Counter	04713	MC14040BCP	117143	2
	U2	Same as U1				
	VR1	DIODE, Zener	04713	1N4733	720388	2
	VR2	Same as VR1				
		CONNECTOR, Female	27264	02-06-2103	10282202	4
		SPACER, Swage 4-40 x 5	06540	9537B-B-0440-3A	100024	2
		SPACER, Swage 4-40 x 375	71279	1300-12	100973	2
		PAD, Transistor for Q1	08289	501-000-D	100666	1
		WASHER, Dome	04713	B52200F003	103351	2
		BOARD, PC	52542	073675	073675	1

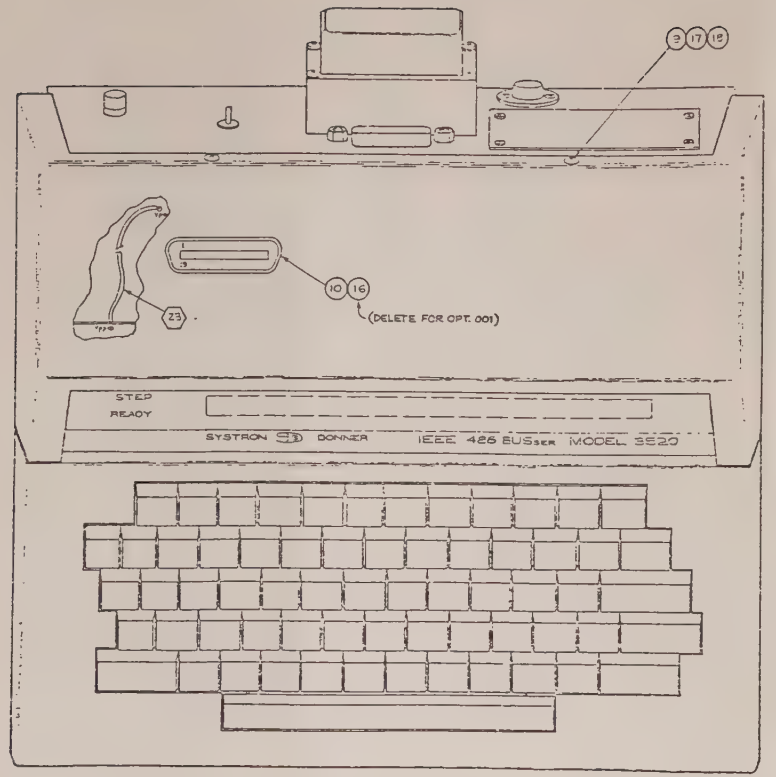
CHAPTER 4
DRAWINGS

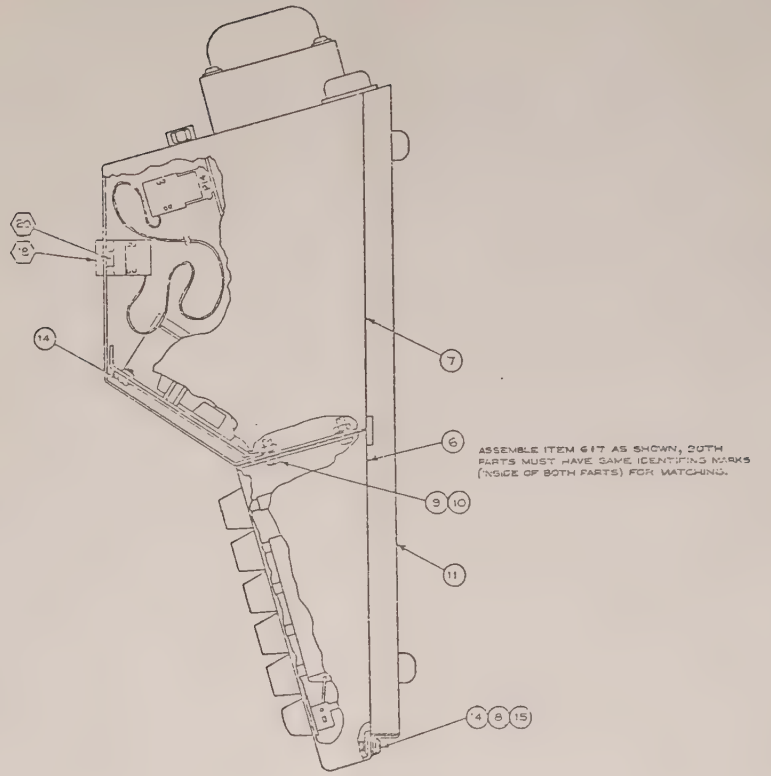
4.1 INTRODUCTION

This chapter contains final assembly and schematic drawings for the Model 3520 Instrumentation Controller. The parts list contained in Chapter 3 relate to the reference designation callouts on the diagrams. Table 4-1 provides a list by figure number of the drawings, with title, drawing number and manual page number.

TABLE 4-1 LIST OF DRAWINGS

Figure No.	Drawing Title	Drawing No.	Page No.
4-1	Model 3520 Final Assembly	07362001	4-3
4-2	Model 3520 Chassis Assembly	07362101	4-4
4-3	A1, Power Supply Assembly	07363401	4-5
4-4	A1, Power Supply Schematic	7-07363401	4-5
4-5	A2, Keyboard PC Assembly	07364001	4-7
4-6	A2, Keyboard Schematic (Sheet 1 of 2)	7-07364001	4-8
4-6	A2, Keyboard Schematic (Sheet 2 of 2)	7-07364001	4-9
4-7	A3, Display PC Assembly	07362301	4-10
4-8	A3, Display Schematic	7-07362301	4-11
4-9	A4, Processor PC Assembly	07362401	4-12
4-10	A4, Processor Schematic	7-07362401	4-13
4-11	Option 001, 2k/4k PROM Module Assembly	07363001	4-14
4-12	Option 001, 2k PROM Module PC Assembly	07369801	4-14
4-13	Option 001, 2k PROM Module Schematic	7-07363001	4-15
4-14	Option 001, 4k PROM Module PC Assembly	07369802	4-16
4-15	Option 001, 4k PROM Module Schematic	7-07363002	4-17
4-16	Control Line Status Indicator Assembly	07366301	4-18
4-17	Control Line Status Indicator PC Assembly	07369901	4-19
4-18	Control Line Status Indicator Schematic	7-073663	4-21
4-19	PROM Eraser Assembly	073701	4-22
4-20	PROM Eraser PC Assembly	07367501	4-22
4-21	PROM Eraser Schematic	7-073701	4-23





⬡ = ITEM Nos OF OPT. 001 P/L 07366801 .

FIGURE 4-1
FINAL ASSEMBLY
#07362001 Rev B

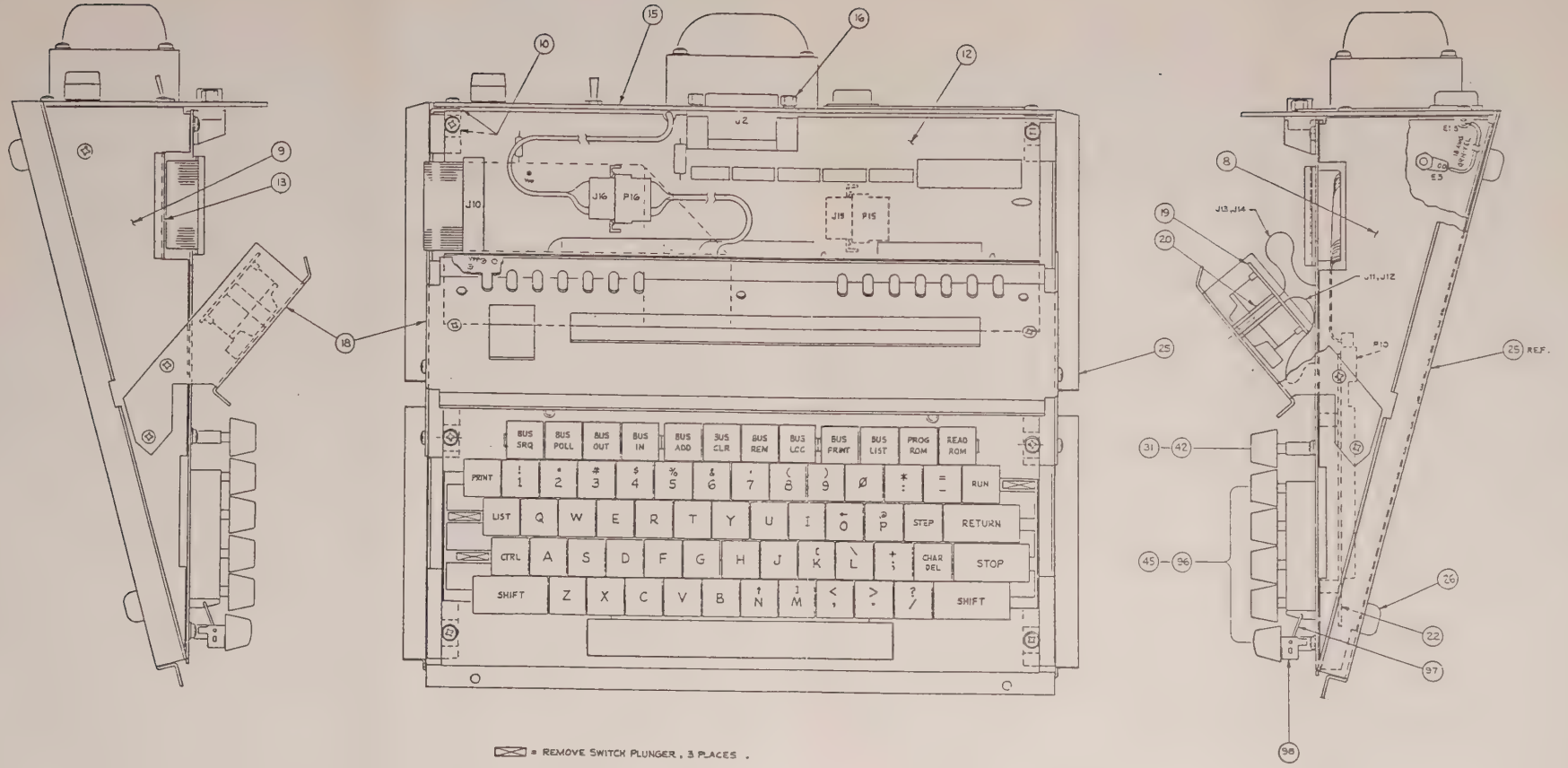


FIGURE 4-2
CHASSIS ASSEMBLY
#07362101 Rev A_2

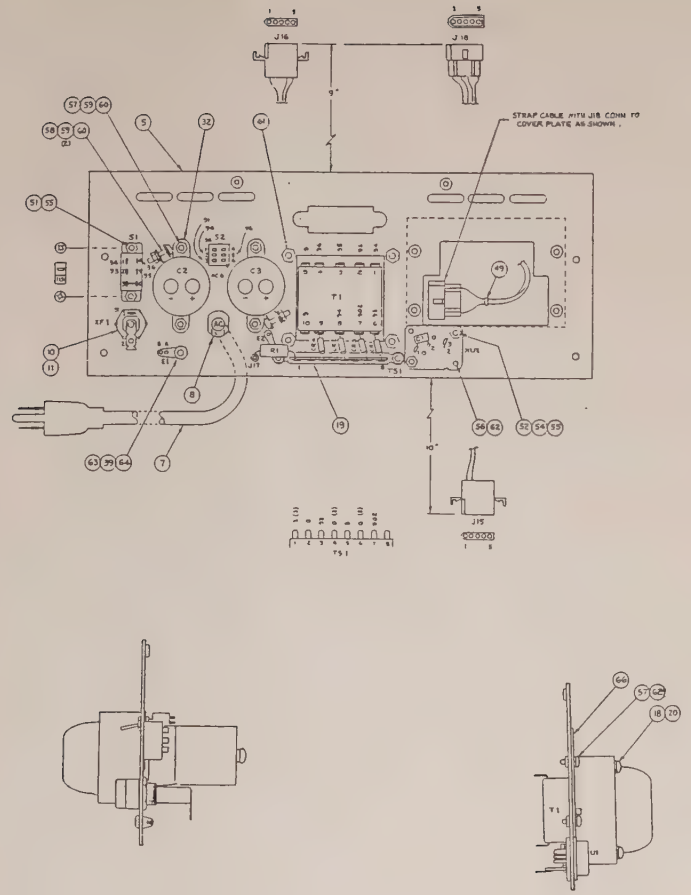


FIGURE 4-3
A1, POWER SUPPLY ASSEMBLY
#07363401 Rev C

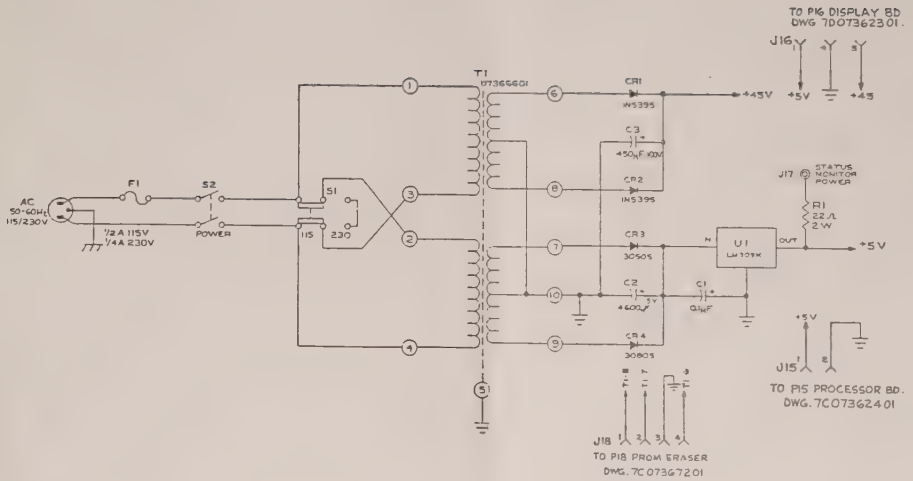


FIGURE 4-4
A1, POWER SUPPLY SCHEMATIC
#7-0736401 Rev B

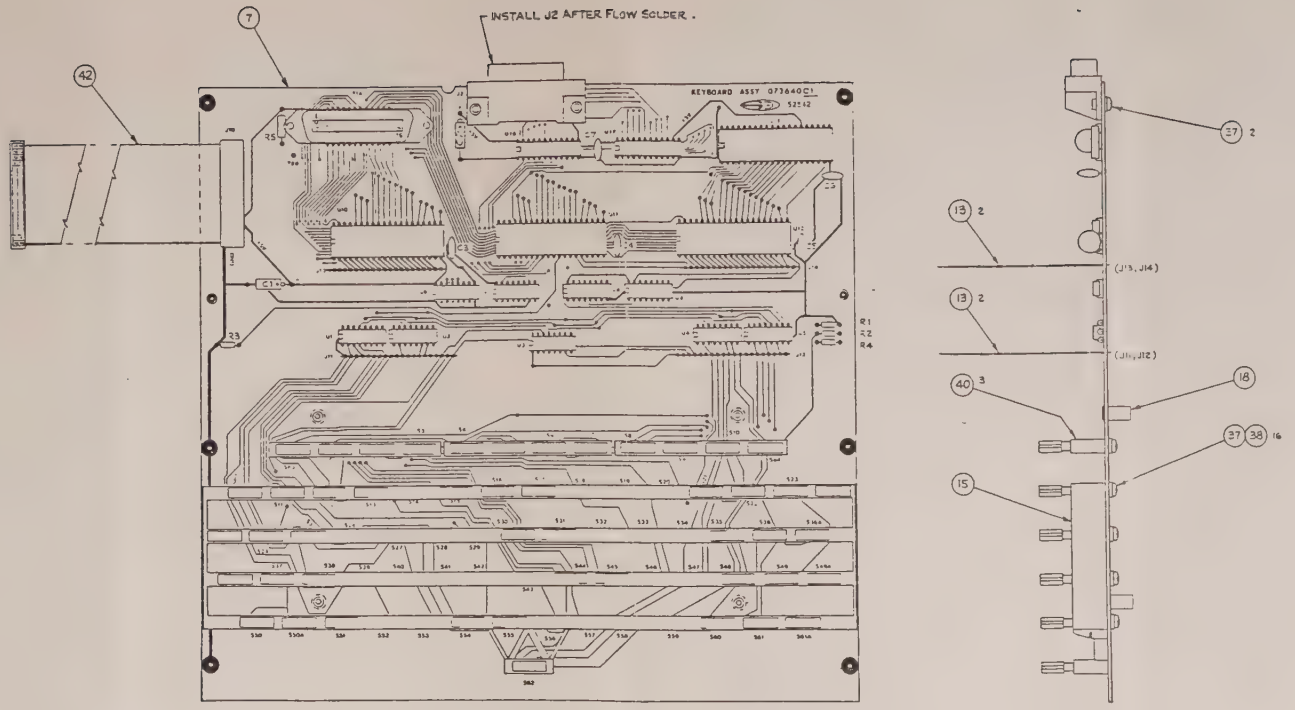


FIGURE 4-5
A2, KEYBOARD PC ASSEMBLY
#07364001 Rev B

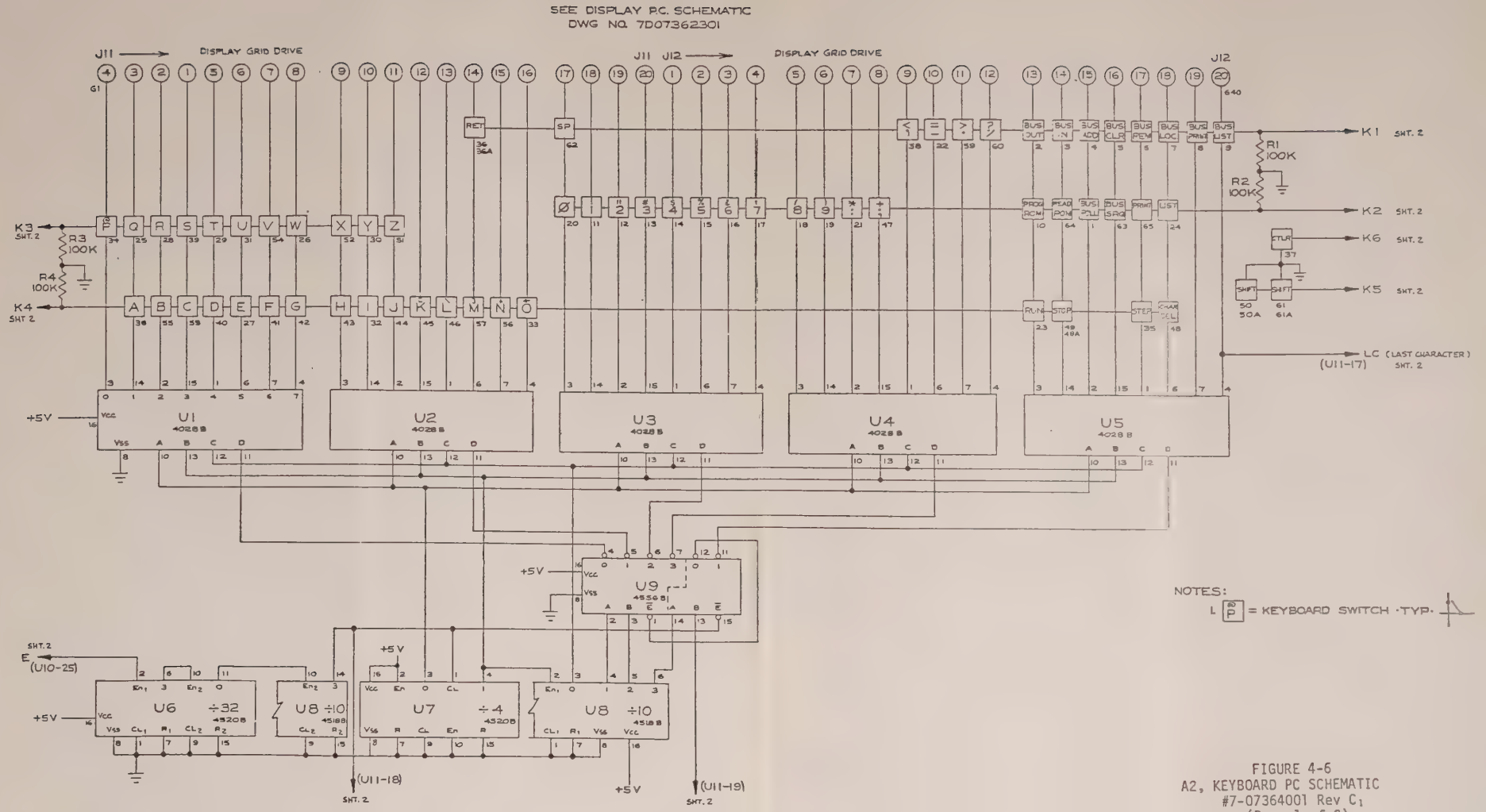
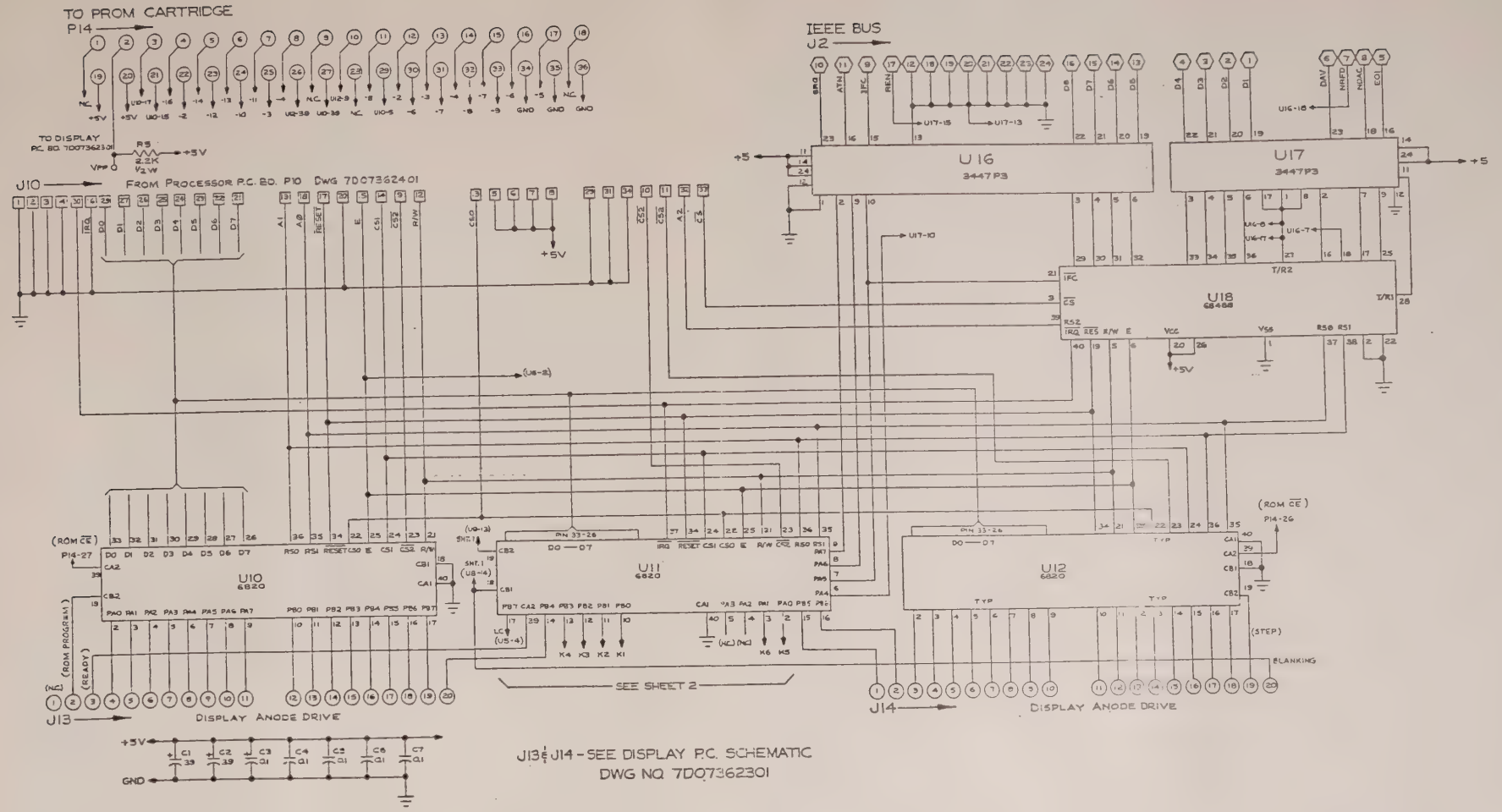


FIGURE 4-6
A2, KEYBOARD PC SCHEMATIC
#7-07364001 Rev C_1
(Page 1 of 2)



J13 & J14 - SEE DISPLAY P.C. SCHEMATIC
DWG NO. 7D0736230I

FIGURE 4-6
A2, KEYBOARD PC SCHEMATIC
#7-07364001 Rev C_1
(Page 2 of 2)

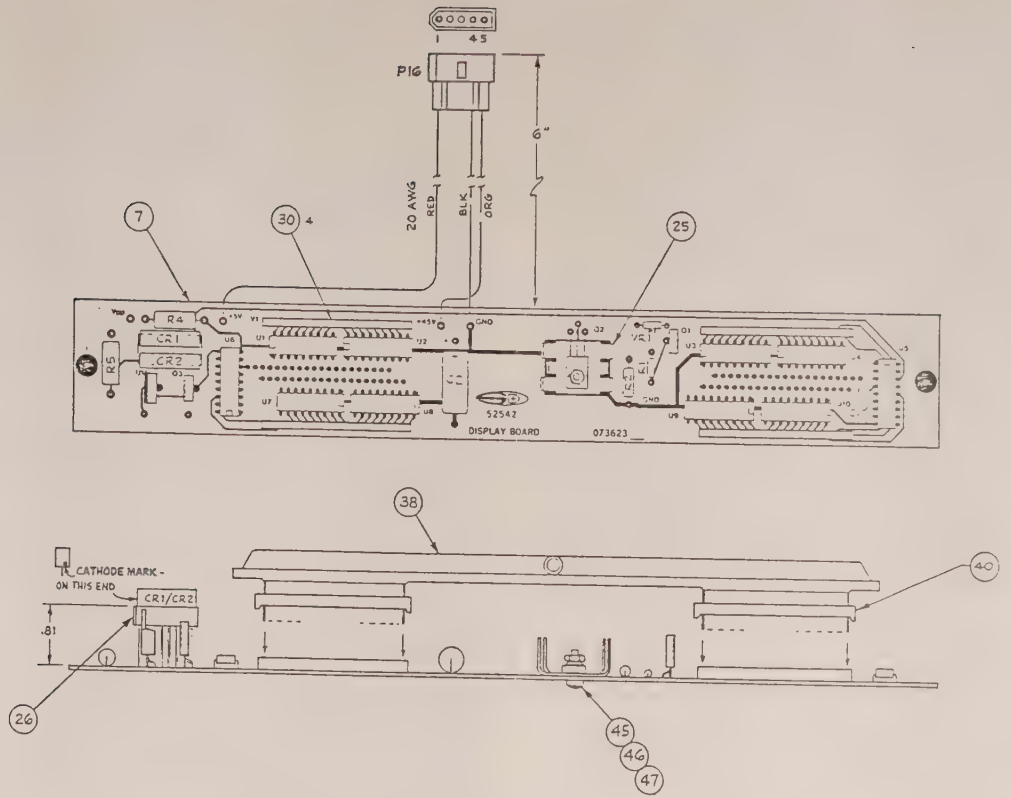
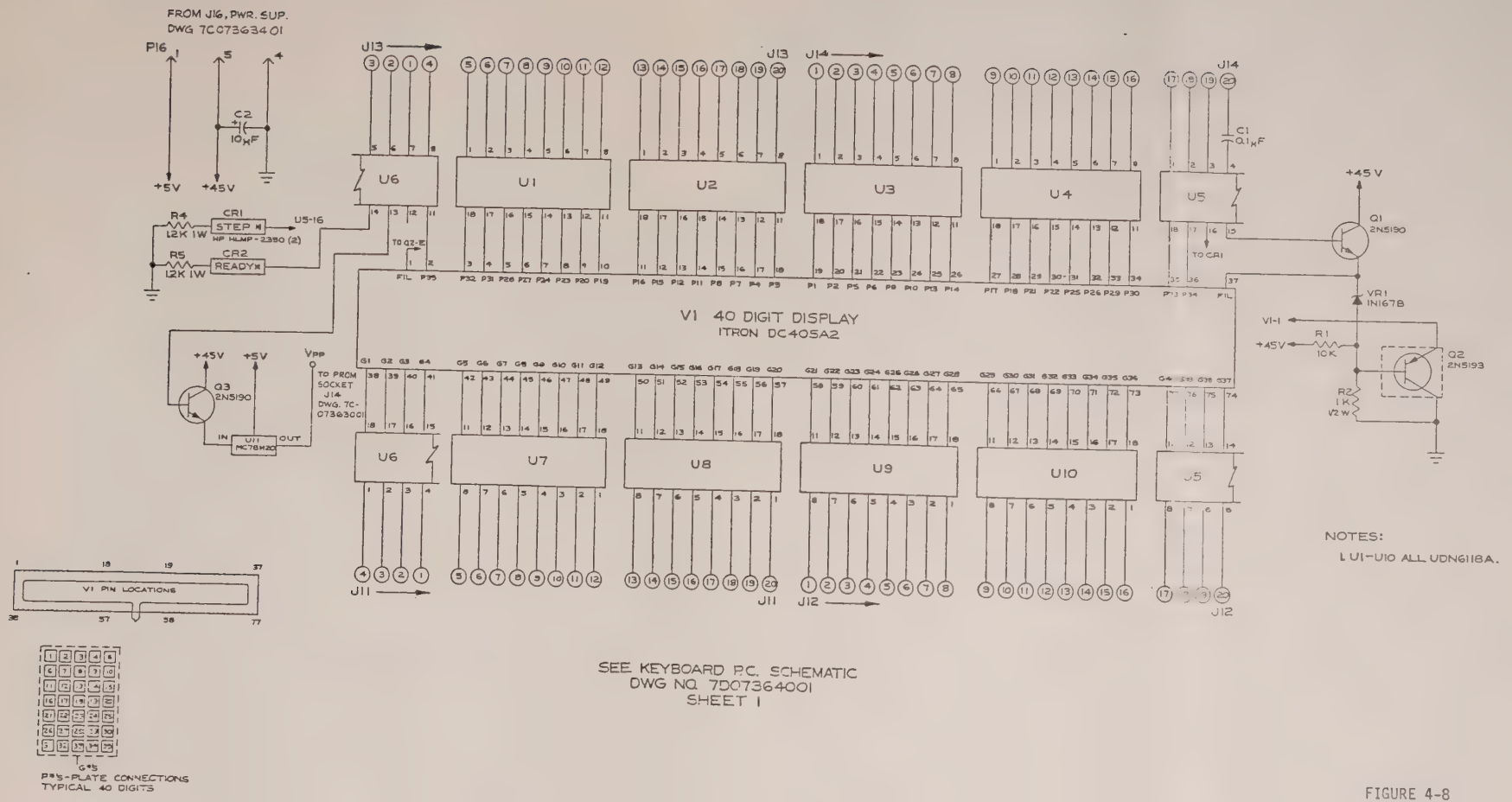


FIGURE 4-7
A3, DISPLAY PC ASSEMBLY
#07362301 Rev B

SEE KEYBOARD P.C. SCHEMATIC
DWG NO. 7D0736400I
SHEET 2



NOTES:
1. U1-U10 ALL UDN6118A.

SEE KEYBOARD P.C. SCHEMATIC
DWG NO. 7D0736400I
SHEET 1

FIGURE 4-8
A3, DISPLAY PC SCHEMATIC
#7-07362301 Rev A_2

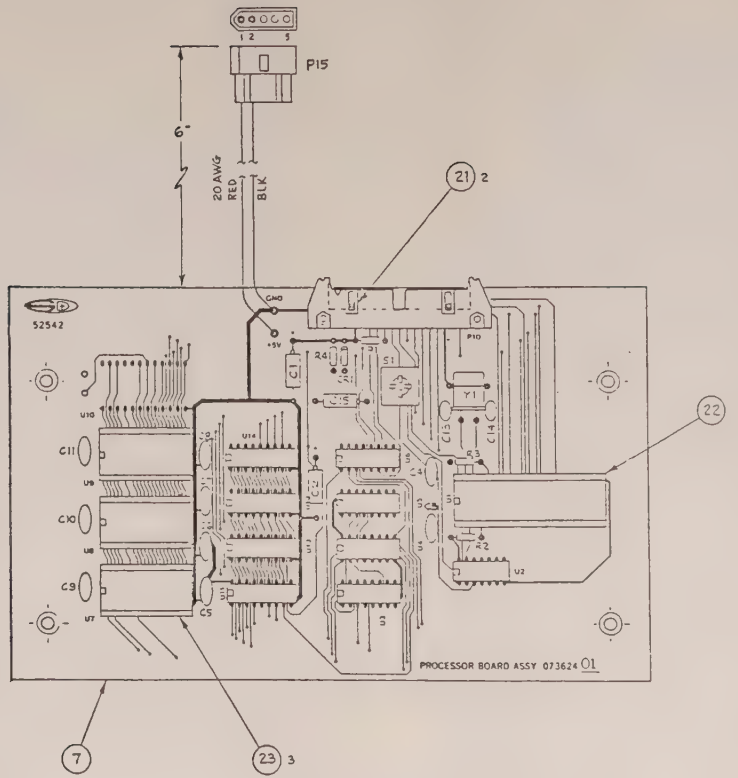
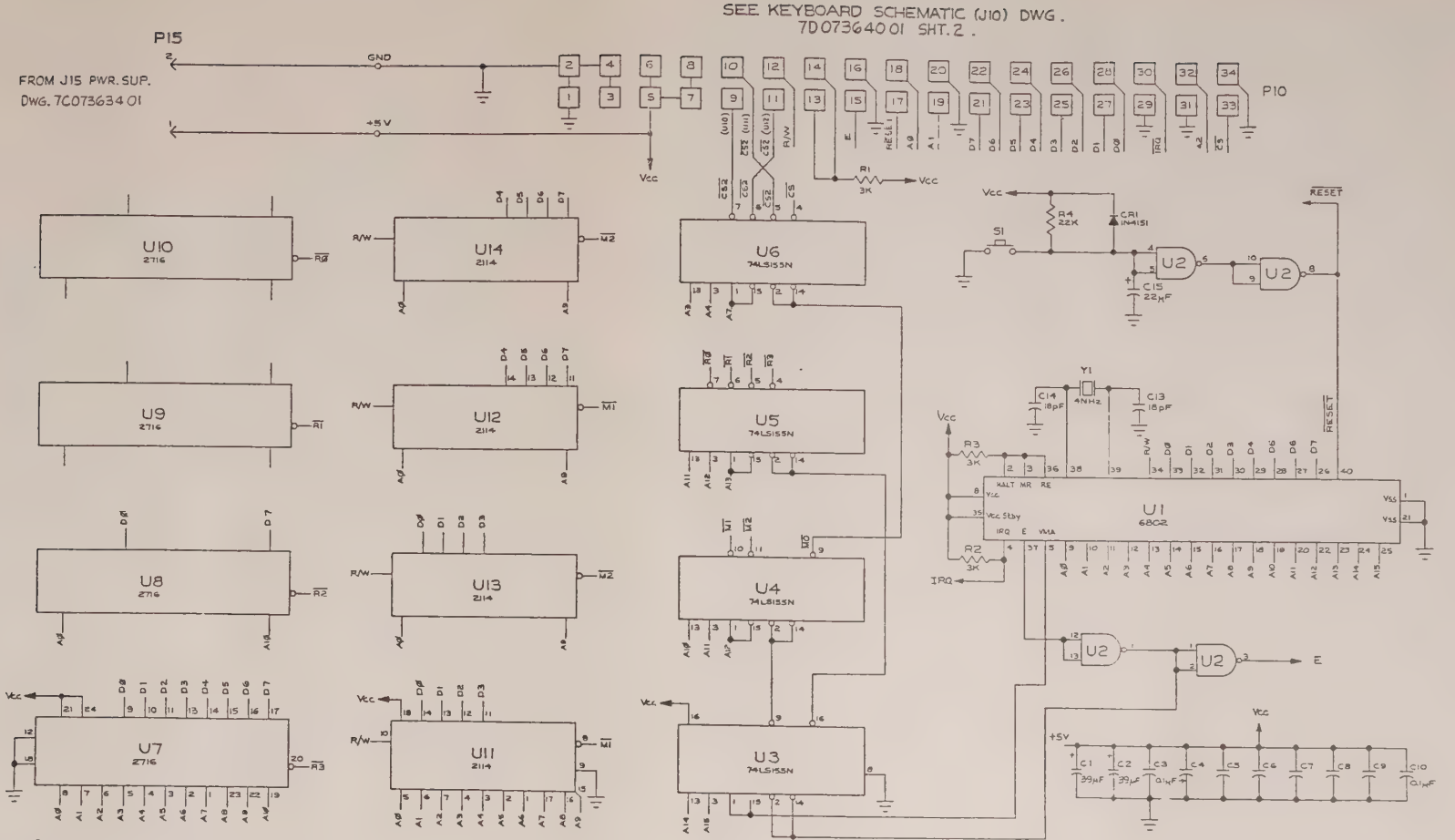


FIGURE 4-9
A4, PROCESSOR PC ASSEMBLY
#07362401 Rev A



NOTES: UNLESS OTHERWISE SPECIFIED.
1. U8, U9, & U10 HAVE SAME PIN PLACEMENT AS U7.
2. U13, U12, & U14 HAVE SAME PIN PLACEMENT AS U11.
3. U2 IS 74132N.

FIGURE 4-10
A4, PROCESSOR PC SCHEMATIC
#7-07362401 Rev A

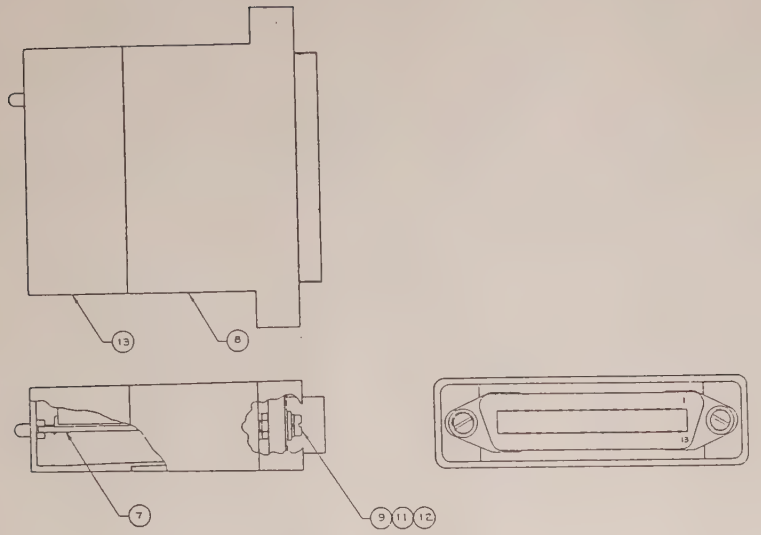


FIGURE 4-11
OPTION 001, 2k/4k PROM MODULE ASSEMBLY
#07363001 Rev A

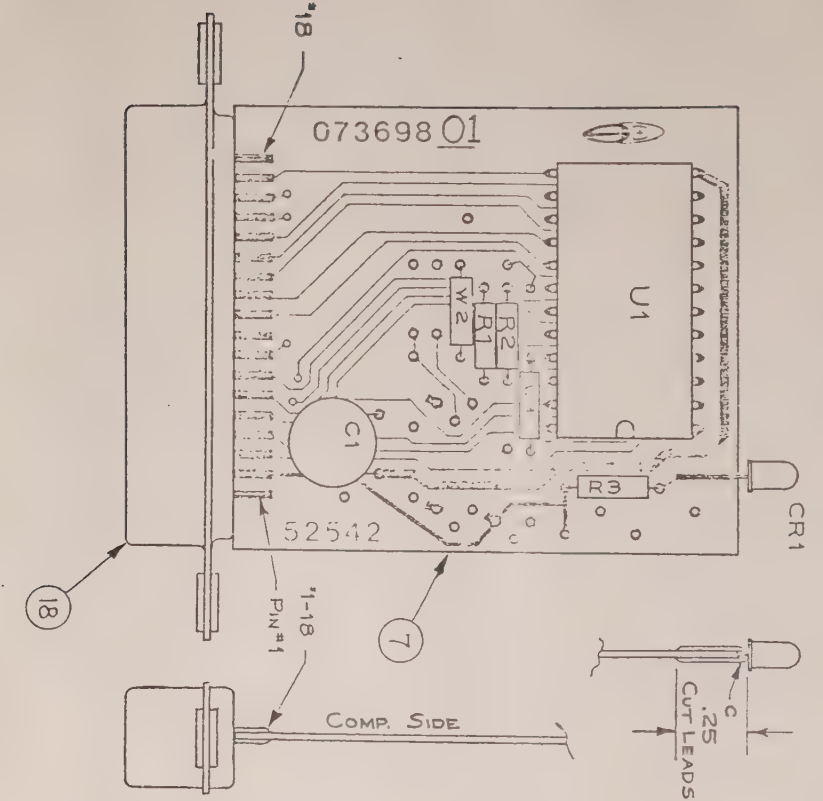


FIGURE 4-12
OPTION 001, 2k PROM MODULE PC ASSEMBLY
#07369801 Rev B

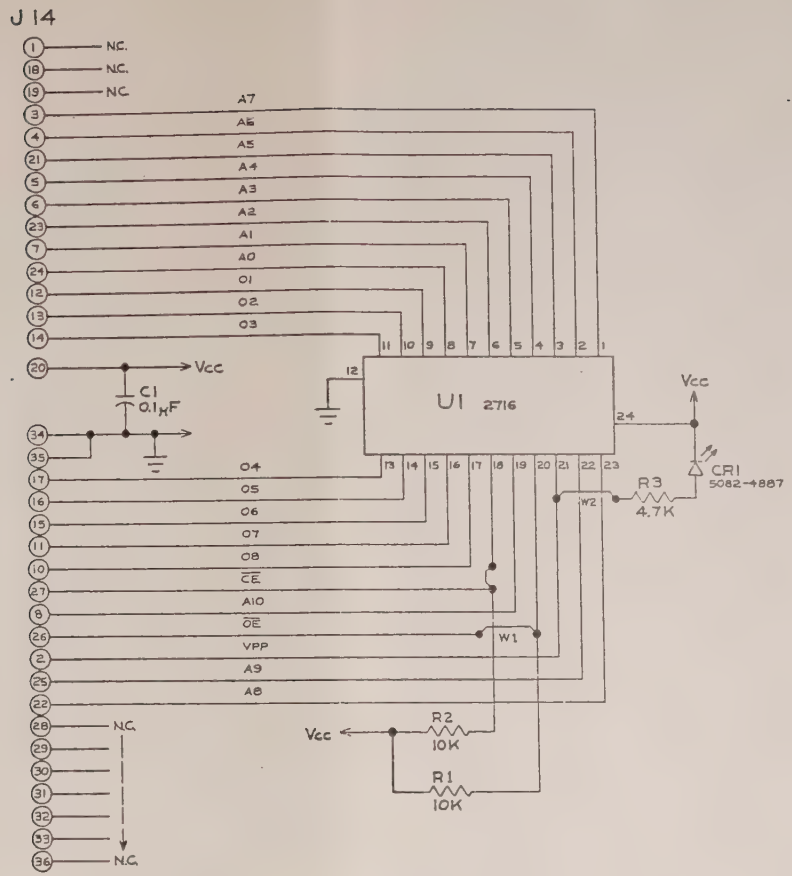


FIGURE 4-13
OPTION 001, 2k PROM MODULE SCHEMATIC
#7-07363001 Rev B

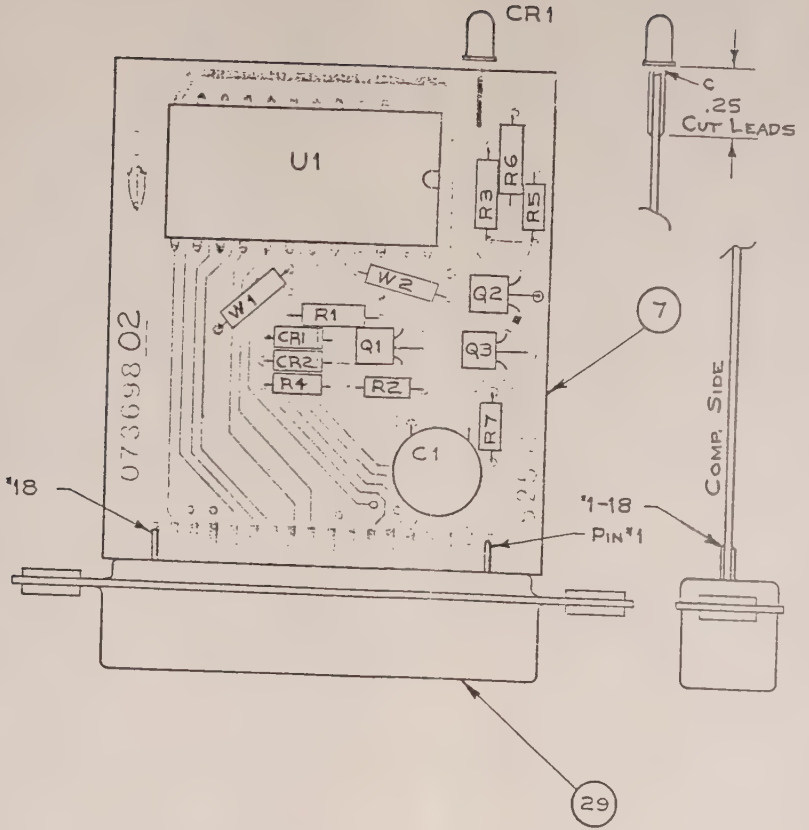


FIGURE 4-14
OPTION 001, 4k PROM MODULE PC ASSEMBLY
#07369802 Rev A

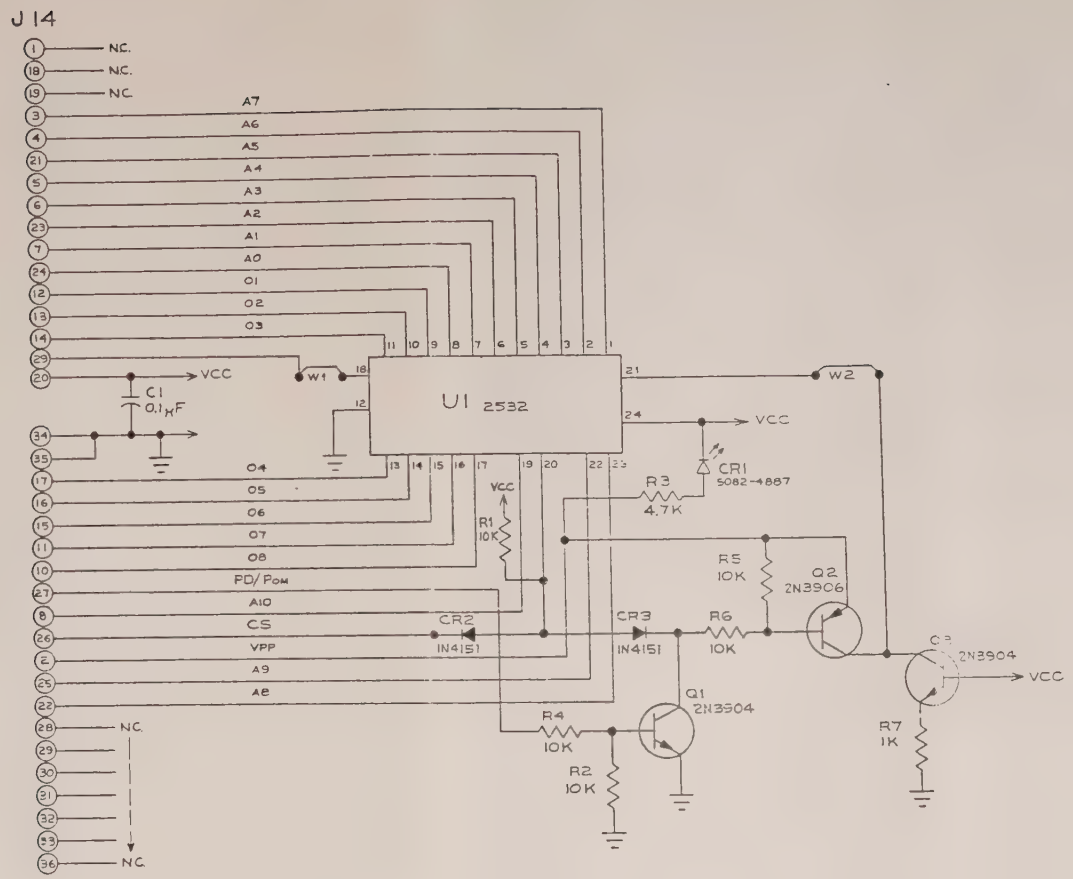


FIGURE 4-15
OPTION 001, 4k PROM MODULE SCHEMATIC
#7-07363002 Rev A

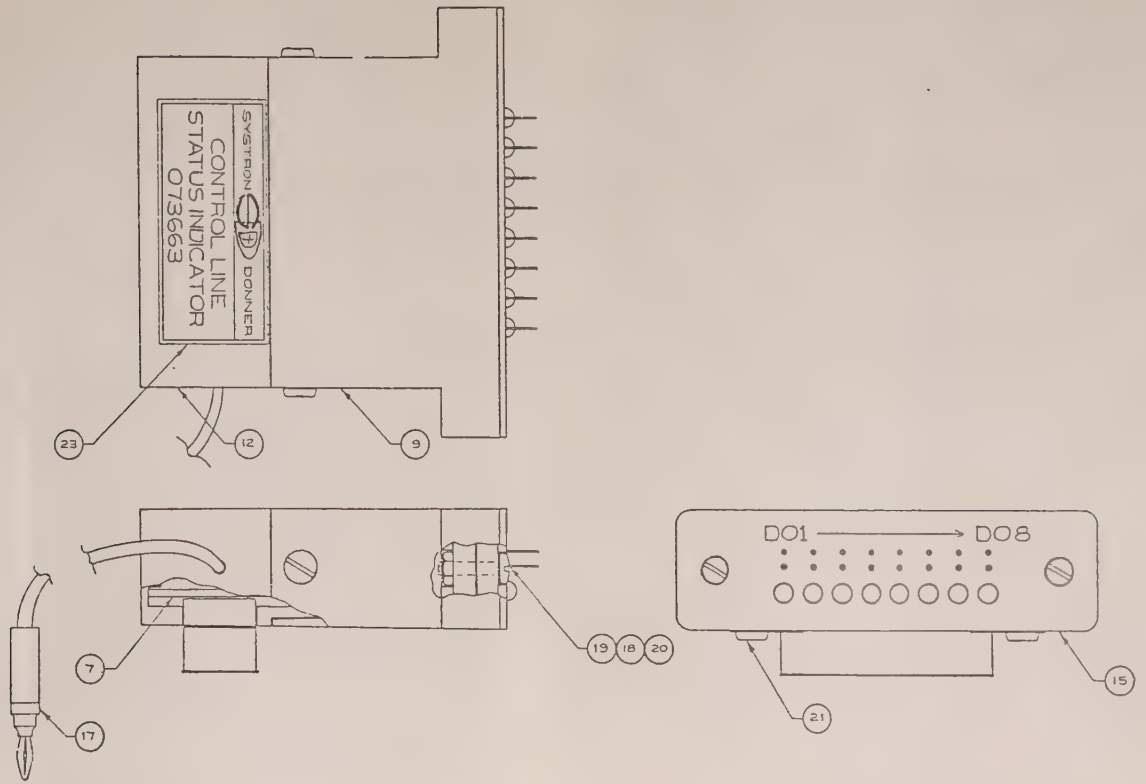


FIGURE 4-16
CONTROL LINE STATUS
INDICATOR ASSEMBLY (ACCESSORY)
#07366301 Rev A_1

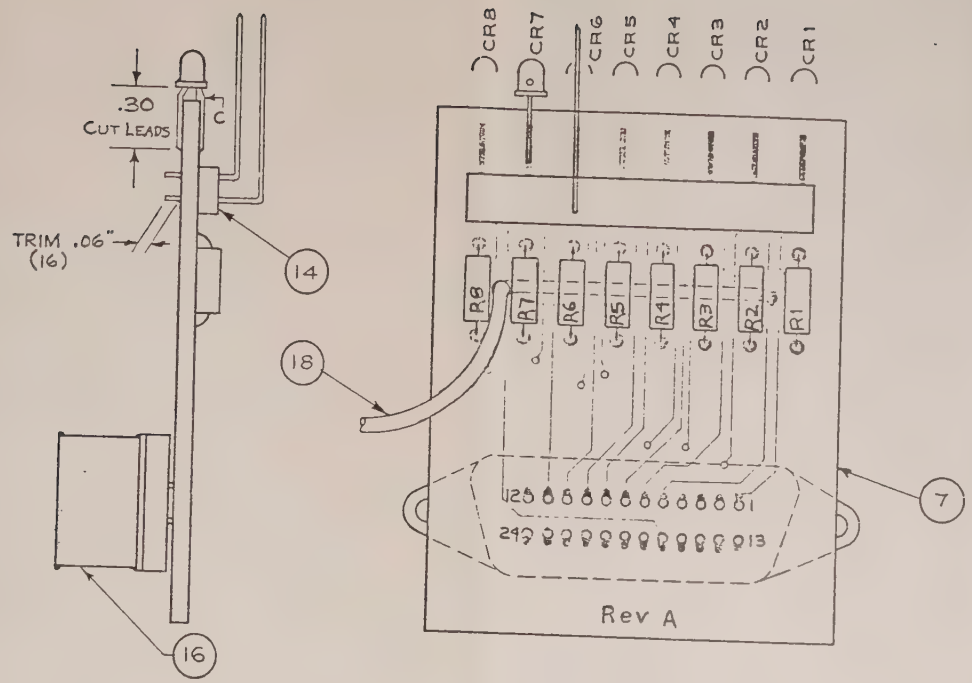


FIGURE 4-17
CONTROL LINE STATUS
INDICATOR PC ASSEMBLY (ACCESSORY)
#07369901 Rev A

3520-5-80

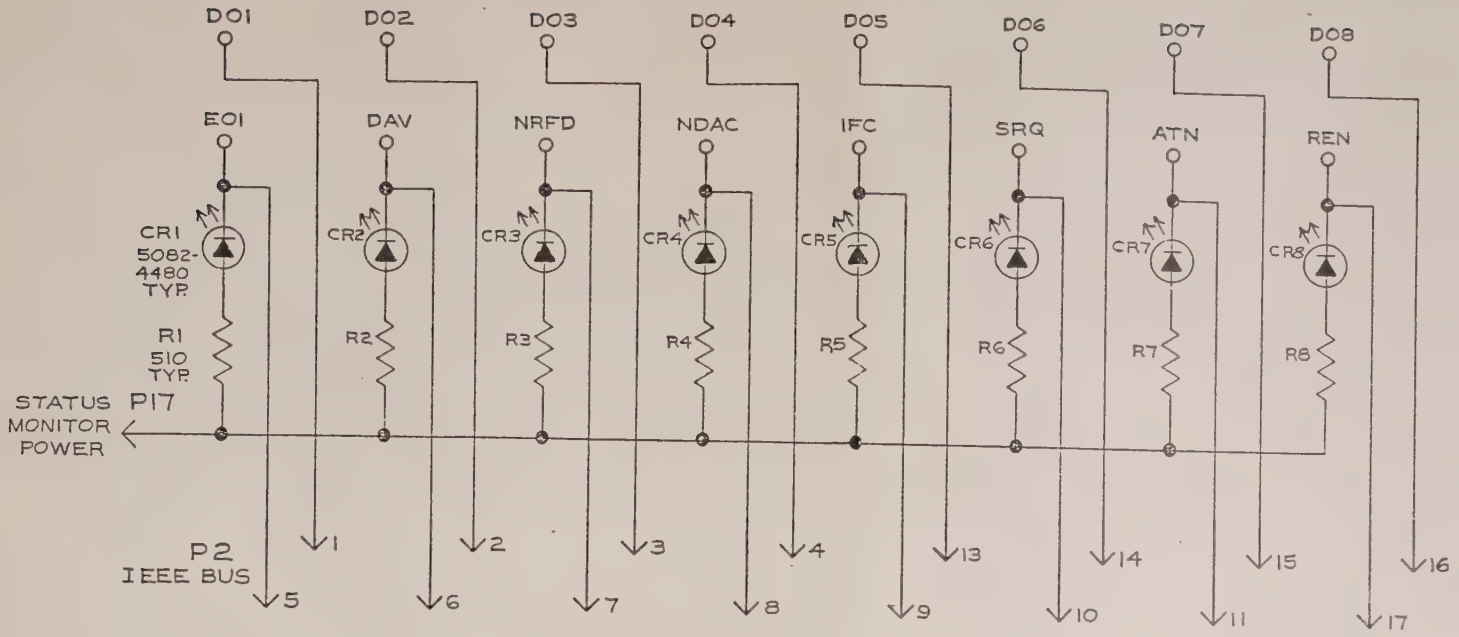


FIGURE 4-18
CONTROL LINE STATUS
INDICATOR SCHEMATIC (ACCESSORY)
#7-073663 Rev A

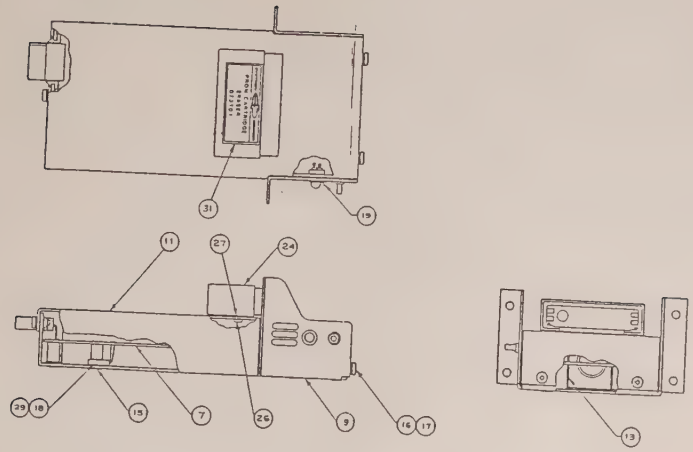


FIGURE 4-19
PROM ERASER ASSEMBLY (ACCESSORY)
#073701 Rev A

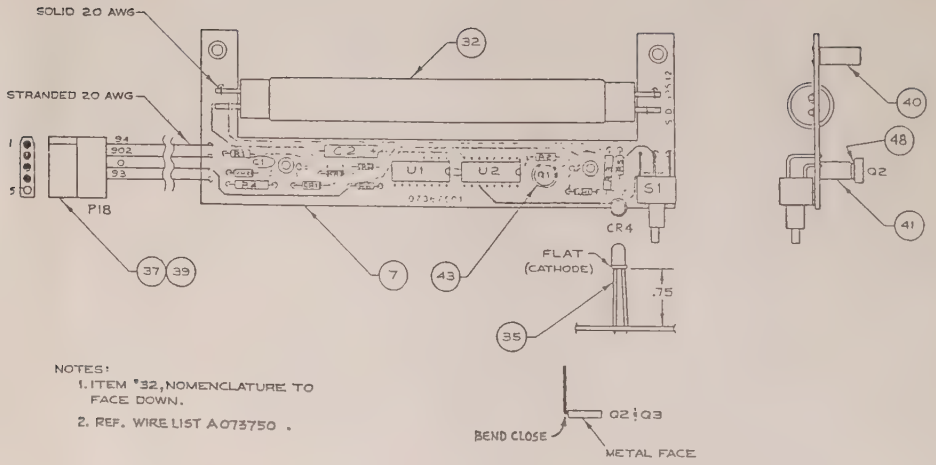
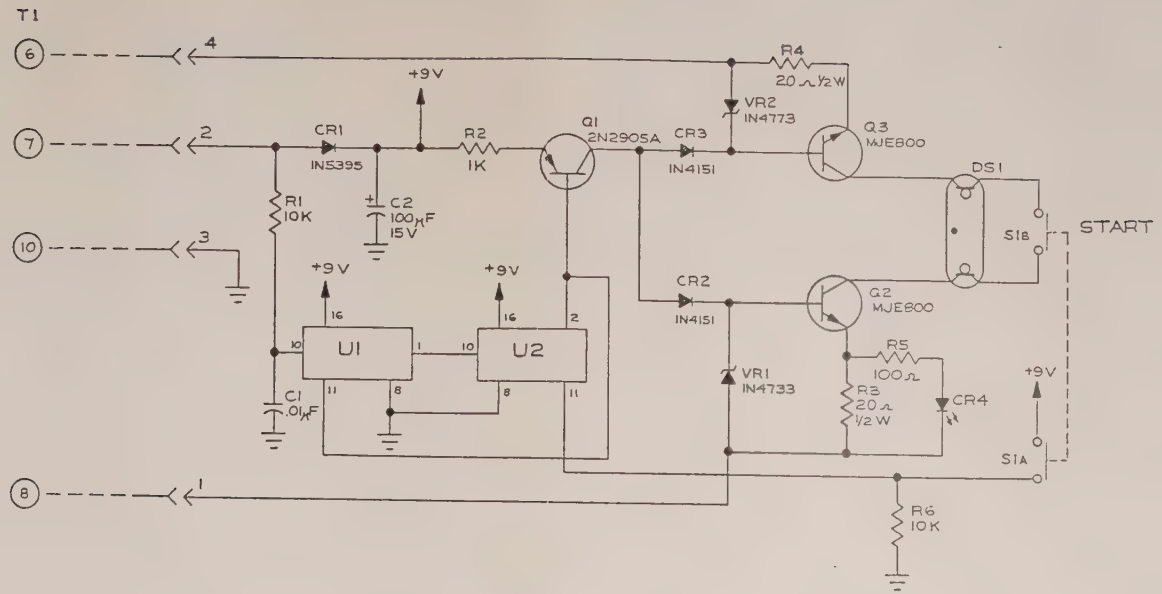


FIGURE 4-20
PROM ERASER PC ASSEMBLY (ACCESSORY)
#07367501 Rev B



NOTES:

1. U1 & U2 ARE MC14040BCP.
2. DANGER : DO NOT EXPOSE EYES TO LAMP.

FIGURE 4-21
PROM ERASER SCHEMATIC (ACCESSORY)
#7-073701 Rev A

TABLE APPENDIX A. BINARY AND DECIMAL CODE TO ASCII CHARACTER

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	DECIMAL	ASCII
0	0	0	0	0	0	1	1	SOH
0	0	0	1	0	1	0	10	LF
0	0	0	1	1	0	1	13	CR
0	1	0	0	0	0	0	32	SP
0	1	0	0	0	0	1	33	!
0	1	0	0	0	1	0	34	"
0	1	0	0	0	1	1	35	#
0	1	0	0	1	0	0	36	$
0	1	0	0	1	0	1	37	%
0	1	0	0	1	1	0	38	&
0	1	0	0	1	1	1	39	'
0	1	0	1	0	0	0	40	(
0	1	0	1	0	0	1	41	)
0	1	0	1	0	1	0	42	*
0	1	0	1	0	1	1	43	+
0	1	0	1	1	0	0	44	,
0	1	0	1	1	0	1	45	-
0	1	0	1	1	1	0	46	.
0	1	0	1	1	1	1	47	/
0	1	1	0	0	0	0	48	0
0	1	1	0	0	0	1	49	1
0	1	1	0	0	1	0	50	2
0	1	1	0	0	1	1	51	3
0	1	1	0	1	0	0	52	4
0	1	1	0	1	0	1	53	5
0	1	1	0	1	1	0	54	6
0	1	1	0	1	1	1	55	7
0	1	1	1	0	0	0	56	8
0	1	1	1	0	0	1	57	9
0	1	1	1	0	1	0	58	:
0	1	1	1	0	1	1	59	;
0	1	1	1	1	0	0	60	<
0	1	1	1	1	0	1	61	=
0	1	1	1	1	1	0	62	>
0	1	1	1	1	1	1	63	?

TABLE APPENDIX A. BINARY AND DECIMAL CODE TO ASCII CHARACTER (Cont'd)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	DECIMAL	ASCII
1	0	0	0	0	0	0	64	@
1	0	0	0	0	0	1	65	A
1	0	0	0	0	1	0	66	B
1	0	0	0	0	1	1	67	C
1	0	0	0	1	0	0	68	D
1	0	0	0	1	0	1	69	E
1	0	0	0	1	1	0	70	F
1	0	0	0	1	1	1	71	G
1	0	0	1	0	0	0	72	H
1	0	0	1	0	0	1	73	I
1	0	0	1	0	1	0	74	J
1	0	0	1	0	1	1	75	K
1	0	0	1	1	0	0	76	L
1	0	0	1	1	0	1	77	M
1	0	0	1	1	1	0	78	N
1	0	0	1	1	1	1	79	O
1	0	1	0	0	0	0	80	P
1	0	1	0	0	0	1	81	Q
1	0	1	0	0	1	0	82	R
1	0	1	0	0	1	1	83	S
1	0	1	0	1	0	0	84	T
1	0	1	0	1	0	1	85	U
1	0	1	0	1	1	0	86	V
1	0	1	0	1	1	1	87	W
1	0	1	1	0	0	0	88	X
1	0	1	1	0	0	1	89	Y
1	0	1	1	0	1	0	90	Z
1	0	1	1	0	1	1	91	[
1	0	1	1	1	0	0	92	\
1	0	1	1	1	0	1	93	]
1	0	1	1	1	1	0	94	^
1	0	1	1	1	1	1	95	←

(SENT AND RECEIVED WITH ATN=1)

Bits b_4	b_3	b_2	b_1	ROW / COLUMN	0_0	MSG (1)	0_1	MSG	1_0	MSG	1_1	MSG	2_0	MSG	2_1	MSG	3_0	MSG	3_1	MSG
					0		1		2		3		4		5		6		7	
0	0	0	0	0	NUL		DLE		SP	MLA ASSIGNED TO DEVICE	0	MLA ASSIGNED TO DEVICE	@	MTA ASSIGNED TO DEVICE	P	MTA ASSIGNED TO DEVICE		MEANING DEFINED BY PCG CODE	p	MEANING DEFINED BY PCG CODE
0	0	0	1	1	SOH	GTL	DC1	LLO	!		1		A		Q		a		q	
0	0	1	0	2	STX		DC2		"		2		B		R		b		r	
0	0	1	1	3	ETX		DC3		#		3		C		S		c		s	
0	1	0	0	4	EOT	SDC	DC4	DCL	$		4		D		T		d		t	
0	1	0	1	5	ENQ	PPC (3)	NAK	PPU	%		5		E		U		e		u	
0	1	1	0	6	ACK		SYN		&		6		F		V		f		v	
0	1	1	1	7	BEL		ETB		'		7		G		W		g		w	
1	0	0	0	8	BS	GET	CAN	SPE	(		8		H		X		h		x	
1	0	0	1	9	HT	TCT	EM	SPD	)		9		I		Y		i		y	
1	0	1	0	10	LF		SUB		*		:		J		Z		j		z	
1	0	1	1	11	VT		ESC		+		;		K		[		k		{	
1	1	0	0	12	FF		FS		,		<		L		\		l		|	
1	1	0	1	13	CR		GS		-		=		M		]		m		}	
1	1	1	0	14	SO		RS		.		>		N		^		n		~	
1	1	1	1	15	SI		US		/		?	UNL	O		_	UNT	o		DEL	

ADDRESSED COMMAND GROUP (ACG)

UNIVERSAL COMMAND GROUP (UCG)

LISTEN ADDRESS GROUP (LAG) (4)

TALK ADDRESS GROUP (TAG)

PRIMARY COMMAND GROUP (PCG)

SECONDARY COMMAND GROUP (SCG)

NOTES: (1) MSG = INTERFACE MESSAGE

(2) b_1 = DIO1...b_7 = DIO7

(3) REQUIRES SECONDARY COMMAND

(4) DENSE SUBSET (COLUMN 2 THROUGH 5)

APPENDIX B
MULTILINE INTERFACE MESSAGES: ISO CODE REPRESENTATION

APPENDIX B
MULTILINE INTERFACE MESSAGES: ISO-7 BIT CODE REPRESENTATION

www.manualsplus.com
1015987